Video Object Segmentation

WEI Wei

A Thesis Submitted in Partial Fulfillment

of the Requirements for the Degree of Master of Philosophy

in

Electronic Engineering

December 2005

ACKNOWLEDGEMENT

I would like to take this opportunity to express my sincere appreciation to the project supervisor, Prof. Ngan King Ngi, for his kind guidance and patience throughout the research. I am deeply impressed by Prof. Ngan King Ngi's serious attitude to research work. It is very helpful to my further study and work.

In addition, I should thank Dr. Hong Liang Li and Dr. Cen Feng who have contributed a lot of constructive suggestions to the project. It is also my great pleasure to thank and acknowledge fellow graduate students in the VSPC lab who have helped me throughout the project development.

I am eternally indebted to my parents for their love, support and advice, which words cannot describe.

Abstract

Recently, object-based video modeling has been used for applications demanding more flexible and accurate video representation. The MPEG-4 video coding standard considers the 2D shape of moving objects not only for reasons of coding efficiency, but also to provide the user with the so-called content-based functionalities. However, in MPEG-4 the decomposition or spatial-temporal segmentation of a scene into objects is not standardized. Therefore, many object-based segmentation algorithms have been proposed in the literature recently.

These segmentation algorithms use different sets of techniques and result in different performance. Although many approaches try to evaluate image segmentation quantitatively, a universal algorithm for segmenting images and a general criterion for the evaluation of segmentation results do not exist, and most techniques are tailored to particular applications. Our work is to develop new and effective strategies to extract visual objects from video sequences. In this thesis, we will review some important motion segmentation and VOP generation techniques that have been proposed. Then, we will introduce our segmentation algorithm. For automatic extraction, we assume that the object of interest can be characterized by a coherent motion that is distinct from that of the background. Firstly, the k median-clustering algorithm is employed to segment the frame of a video sequence into homogeneous regions based on luminance, chrominance, texture, position and motion information. Then the video objects will be tracked by using the object binary model or region descriptors. We also did some work on stereo disparity

estimation, which uses the edge matching and spatial interpolation algorithms on color information. Disparity information can be an important cue to perform video segmentation in some specific applications, like human extraction in videoconference sequence. The performance of our algorithm is demonstrated by experimental results. Finally, we will make a conclusion and introduce the future work.

Keywords: Content-based coding, MPEG-4, video object planes, disparity estimation

摘要

近年來，由於能更加靈活和準確的表達視頻內容，基於物體的視頻模型得到越來越廣泛的應用。MPEG-4 視頻編碼標準基於物體的二維形狀進行編碼，不僅是為了其編碼的有效性，更因為這樣能夠提供給用戶被稱為基於視頻內容的操作功能。然而 MPEG-4 將整個場景分解為物體並沒有一定的標準。 所以，在近年的文獻中出現許多關於物體分割算法的討論。

這些算法使用不同的技術並且應用於不同的領域。雖然存在很多評價圖像分割的方法，但是通用的圖像分割算法和標準的評估方法並不存在，並且大多數的算法都是適應於特別的應用。我們的工作是提供新的有效的分割策略將物體從視頻序列中提取出來。 在這篇論文中，我們首先回顧一些關於運動物體分割經典方法，然後介紹我們的工作。 由於要做到自動提取物體，我們假設感興趣的物體存在與背景不一樣的運動。 首先，我們使用 k 中值聚類的方法運用亮度，色度，紋理，位置和運動等信息，將視頻序列的每一幀劃分為不同的區域。 然後我們對運動物體建立二維模型或者是使用區域描述的方法將物體提取出來。我們也在立體對偶照片的深度估計方面做了一些工作， 我們使用邊界匹配和運用色度插值的方法得到深度信息，並將它運用於視頻分割中，例如視頻會議的人物提取。實驗結果證實我們算法的有效性。

List of Abbreviations

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
CDM	Change Detection Mask
DFD	Displaced Frame Difference
EM	Expectation and maximization
FD	Frame Difference
FSM	Face Saliency Map
HOS	Higher Order Statistic
HT	Hough transform
IMC	Independently Moving Components
MAP	Maximum a posterior
MPEG	Moving Picture Experts Group
MRF	Markov random field
MSE	Median Squared Error
QCIF	Common Intermediate Format
SAD	Sum of Absolute Difference
VO	Video Object
VOP	Video Object Plane

TABLE OF CONTENTS

TABLE OF FIGURES

Chapter 1
Introduction

1.1 Overview of Content-based Video Standard

Moving Pictures Experts Group (MPEG) is a working group of ISO/IEC in charge of the development of international standards for compression, decompression, processing, and coded representation of moving pictures, audio, and their combination.

So far, MPEG has produced MPEG-1 [MP93], the standard for storage and retrieval of moving pictures and audio on storage media (approved November 1992), MPEG-2 [MP95], the standard for digital television (approved November 1994). MPEG-1 and MPEG-2 have provided the foundation for the digital representation of the audiovisual information, the successful convergence and implementation of which have become a catalyst for propelling the new digital consumer markets such as Video CD, Digital TV, DVD, and DBS. Success of digital television, interactive graphics applications and interactive multimedia encouraged MPEG group to design the MPEG-4 [TS97, RS98, MP98] and MPEG-7 [BS01, SF01], which allows the user to interact with the objects in the scene within the limits set by the author.

The MPEG-4 standard enables content-based functionalities by introducing the concept of video object planes (VOP). In contrast to classical video standards, MPEG-4 considers video sequences as compositions of various audiovisual objects, each of which is semantically meaningful to the viewer (for example persons, trees, houses, cars). With separately encoded objects and by specifying its position in a two- or three- dimensional

space, the decoder is able to reconstruct the original scene. A major strength of this object-oriented representation is that audio and video can be easily manipulated. The following Figure 1.1 describes general block diagram of MPEG-4 video.

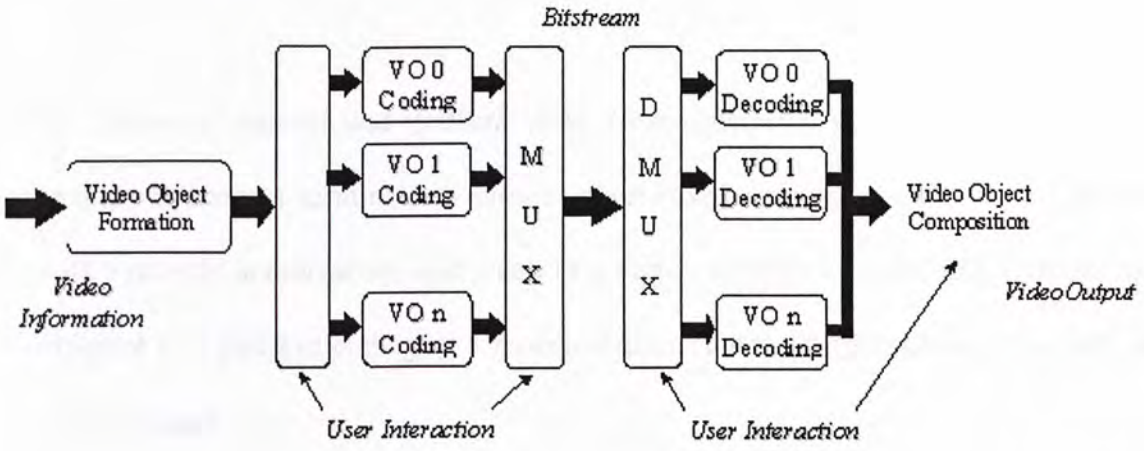


Figure 1.1 Block diagram of MPEG-4 video

The MPEG-4 visual standard consists of a set of tools that enables applications by supporting several classes of functionalities. The most important features covered by MPEG-4 standard can be clustered into four categories and summarized as follows:

- *Universal accessibility and robustness in error prone environments.* Multimedia audiovisual data need to be transmitted and accessed in heterogeneous network environments, possibly under severe error conditions (e.g., mobile channels). Although the MPEG-4 standards will be network (physical layer) independent, the algorithms and tools for coding audiovisual data are designed with awareness of network peculiarities.

- *Content-based interactivity*: Coding and representing video objects rather than video frames enables content-based applications. It is one of the most important novelties offered by MPEG-4. Based on efficient representation of objects, object manipulation, bit stream editing, and object-based scalability allow new levels of content interactivity.

- *Coding of natural and synthetic data.* Next-generation graphics processors will enable multimedia terminals to present pixel-based audio and video data together with synthetic audio/speech and video in a highly flexible way. MPEG-4 assists the efficient and flexible coding and representation of natural (pixel-based) as well as synthetic data.

- *Compression efficiency.* The storage and transmission of audiovisual data requires a high coding efficiency, meaning a good quality of the reconstructed data. Improved coding efficiency, in particular at very low bit rates (<64kbit/s,) is an important functionality to be supported by the MPEG-4 video standard.

In addition to standard MPEG-1 or MPEG-2 like provisions for efficient coding of conventional image or audio sequences, MPEG-4 enables an efficient coded representation of the audio and video data that can be "content-based," with the aim of using and presenting the data in a highly flexible way [LC97, TS97, ST97]. In particular MPEG-4 allows the access and manipulation of audiovisual objects in the compressed domain at the coded data level, to assist future multimedia database access applications

such as the flexible presentation of image or audio content in the World-Wide Web, computer games, and related applications.

Compared with MPEG-4, the MPEG-7 standard, formally named "Multimedia Content Description Interface", provides a rich set of standardized tools to describe multimedia content. Both human users and automatic systems that process audiovisual information are within the scope of MPEG-7. MPEG-7 offers a comprehensive set of audiovisual Description Tools (the metadata elements and their structure and relationships, that are defined by the standard in the form of Descriptors and Description Schemes) to create descriptions (i.e., a set of instantiated Description Schemes and their corresponding Descriptors), which will form the basis for applications enabling the needed effective and efficient access (search, filtering and browsing) to multimedia content. This is a challenging task given the broad spectrum of requirements and targeted multimedia applications, and the broad number of audiovisual features of importance in such context.

1.2 Video Object Segmentation

1.2.1 Video Object Plane (VOP)

In object-based coding, the Video Object (VO) corresponds to entities in the bitstream that the user can access and manipulate (cut, paste…) [HT]. Instances of Video Object in given time are called Video Object Plane (VOP). The video frames are defined in terms of layers of VOP. The encoder sends together with the VOP, composition

information (using composition layer and syntax) to indicate where and when each VOP is to be displayed. At the decoder side, the user may be allowed to change the composition of the scene displayed by interacting on the composition information.

The VOP can be a semantic object in the scene: it is made of Y, Cb, Cr components plus shape information. Shape information is used to mask the background, and help to identify object boundaries. The shape information is used to form a VOP. The VOP is formed by first drawing the tightest rectangle around the object. The rectangle is then extended to a bounding rectangle that contains a multiple of macroblocks as shown in Figure 1.2. This ensures that the VOP contains a minimum number of macroblocks to represent the object.

Figure 1.2 VOP Formation

1.2.2 Object Segmentation

If VOPs are not available, then video frames need to be segmented into objects and a VOP to be derived for each one. In general, segmentation consists of extracting image

regions of similar properties such as brightness, color or texture. These regions are then used as masks to extract the objects of interest from the image.

Object segmentation in computer vision consists of the extraction of the shape of the physical objects projected onto the image plane, ignoring edges due to texture inside the object borders [MG99]. This extremely difficult image processing task differs from the most basic segmentation problems usually formulated as separation of image areas containing pixels with similar intensity, in the objective of the task itself. While the result of general segmentation can be a large number of irregular segmentations (based only on intensity similarity), object segmentation tries to recognize the shapes of complete physical objects present in the scene. It is intuitively clear that this more general segmentation cannot be carried out without any additional information about the structure or dynamics of the scene. In this context most approaches for object segmentation can be included in two broad classes. The first one concerns methods for extraction of object masks by means of multi-view image analysis on sequences taken from different perspectives, e.g., stereoscopic images, exploiting the 3D structure of the scene. The second is motion-based segmentation when only monoscopic sequences are available. In the latter case the dynamics of objects present in the scene is exploited in order to group pixels that undergo the same or similar motion. Because most natural scenes consist of locally rigid objects and moving objects deform continuously in time, it is expected that connected image regions with similar motion belong to a single object.

1.3 Problems of Video Object Segmentation

In order to carry out object based video coding, video sequence should be segmented in terms of meaningful semantic objects and has low computation. If there is only one VOP consisting of the whole rectangular frame, no explicit segmentation is necessary. However, in practice, one can hardly find a video sequence that contains a single object. There are usually at least two objects: a stationary background and a moving foreground. Moreover, there are often more than one foreground object. So it is necessary for us to partition a video sequence into VOP's by means of segmentation algorithm.

Decomposing video sequences into VOP's is very difficult in many cases. An intrinsic problem of VOP generation is that objects of interest are not homogeneous with respect to low-level features, such as color, intensity, or optical flow. Instead, VOP segmentation involves higher-level semantic concepts. Hence, conventional low-level segmentation algorithms will fail to obtain meaningful partitions. At the moment, there is not any algorithm that can automatically perform VOP segmentation accurately and reliably for generic video sequences. The main difficulty is to formulate semantic concepts in a form suitable for a segmentation algorithm.

1.4 Objective of the research work

The objective of this research work is to develop new and effective strategies to extract visual objects from video sequences for content-based video coding. The suitability of various video segmentation approaches for extracting semantically meaningful objects will be investigated first, and then potentially useful methods will be analyzed with regard to their strengths and weaknesses. Addressing the shortcomings of

current video segmentation techniques, new methods, which use the combination of many features including spatial and temporal, and require minimal human interactions, will be developed to segment semantic objects in the various kinds of video sequences.

1.5 Organization of This Thesis

- Chapter 2 provides a brief review on video segmentation.
- Chapter 3 describes the proposed video object segmentation system.
- Chapter 4 presents the disparity estimation algorithm and its application in video object segmentation.
- The final chapter gives a conclusion of the work done in this research. Future works that could be extended are also discussed.

1.6 Notes on Publication

Parts of the work reported in this thesis have been reported in the following papers:

- W. Wei and K. N. Ngan," Disparity Estimation with Edge-based Matching and Interpolation", ISPACS, Hong Kong, Dec 2005
- W. Wei and K. N. Ngan, "Integration of motion and image features for automatic video object segmentation," IEEE International Conference on Image Processing (ICIP), Singapore, Oct. 2004.
- W. Wei and K. N. Ngan, "Multiple feature clustering algorithm for automatic video object segmentation," IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Canada, May 2004.

- W. Wei and K. N. Ngan, "Automatic Video Object Segmentation for MPEG-4," SPIE Visual Communications and Image Processing (VCIP), Switzerland, Jul. 2003
- W. Wei and K. N. Ngan, "Automatic video object segmentation with multiple features ", submitted to IEEE Trans. on Circuits and Systems for Video Technology

Chapter 2
Literature Review

2.1 What is segmentation?

The word "segmentation" has a meaning that depends to a large extend on the application and the context in which it is used. It is the process of dividing an image into regions of interest [BE90]. In this process all the pixels in such a region are given the same label. The basic goal of any segmentation algorithm is to define a partition of the space. In the context of image and video, the space can be temporal (1D), spatial (2-D) or spatio-temporal (3-D). The process can be done manually, automatically, semi-automatically, or using blue screen technology.

2.1.1 Manual Segmentation

Manual segmentation methods include pixel selection, geometrical boundary selection, and tracing. The rules representing the semantic information are applied directly by the user. This procedure allows a perfect definition of the object boundaries. Spatial accuracy and temporal coherence are therefore guaranteed. However, the procedure is very time consuming, given normal image resolutions of 176 $\times$352 or greater, selection of individual pixels is clearly impractical and rarely used, especially if the boundary can be approximated by simple polygons or ellipses.

A manual approach is necessary in some cases, such as high quality film production of creation of reference segmentation in order to assess the quality of automatic or semi-automatic extraction techniques [XP00]

2.1.2 Automatic Segmentation

Fully automatic segmentation methods are usually impractical due to image complexity and the variety of image types and interpretations. At the moment, there is no algorithm that can automatically perform VOP segmentation accurately and reliably for generic video sequences. The main difficulty is to formulate the semantic concepts in a form suitable for a segmentation algorithm. So the automatic methods are usually derived for a specific application, or class of applications. A typical example of methods based on a specific set-up of the scene is the *blue screen* approach (chroma-keying). Examples of methods based on *a prior* information are template matching, face detection, and moving object segmentation. However, in most cases, this a priori information is not available to the computer prior to segmentation without user interaction. In addition, low contrast between structures generally causes even most "robust" automatic algorithms to fail.

Chroma-keying — the blue/green technology uses chroma as cue for segmentation. In the *blue screen* approach to automatic extraction, objects in the real world are filmed in front of a uniformly colored background (usually blue or green). The background is then eliminated by discarding pixels with the known background color. The blue screen approach provides good spatial accuracy and temporal coherence. However, it is not generically applicable, because it requires a specific set-up of the scene. Besides, special

care is required for the lighting of scene, to avoid shadows. This is of limited practical applications.

***A prior* information**—In this approach, some knowledge of objects we want to extract substitutes the knowledge of color of the background of the blue screen approach. Template matching, face detection, and moving object segmentation are typical examples of methods based on *a prior* information. If the shape of the object we want to segment is known *a priori*, *template matching* can be used to implement the semantics. In this case, the extraction method will look for specific object features in terms of geometry. If we want to segment faces of people, color-based segmentation can be used. The *face detection* task will consist in finding the pixels whose spectral characteristics lie in a specific region in the chromaticity diagram [ML04]. For extracting moving objects, *motion information* can be used as semantics. The motion of moving object usually differs from the motion of background and other object. For this reason, many extraction methods make use of motion information in video sequences to automatically extract semantic objects. Extraction strategies based on motion are extensively reviewed in Section 2.3.

2.1.3 Semi-automatic segmentation

Semi-automatic segmentation methods combine the benefits of both manual and automatic segmentation techniques. The principle at the basis of semi-automatic technique is the *interaction* of the user during some stages of the extraction process, where the semantic information is provided directly by the user. By supplying initial

information about video object, the user may guide an otherwise automatic segmentation procedure. By adjusting this information, the user may use the results of the segmentation procedure as increasingly accurate approximations to the actual region. Finally, any remaining errors introduced by automatic segmentation methods may be corrected by manual editing [FG98, CG98, SC98]. Currently, this appears to be the most promising approach unless a very constrained situation is present.

According to the choice that the user makes in the interaction, semi-automatic techniques can be classified as *feature-based*, *contour-based*, or *region-based*. These three approaches may be used either separately or in combination [BF99], thus allowing good flexibility.

Feature-based interaction — In the case of feature-based interaction, the user selects some pixels belonging to the object that exhibit characteristic color/texture properties. These pixels are used as basis for extraction: they are characterized by their features and the remaining pixels are then classified accordingly [EC96]. The advantage of this method is that it is not necessary to demarcate the object precisely. However, there might be problems in the connectivity of the object. A further interaction step is required to overcome this problem.

Contour-based interaction — Instead of selecting a set of points belonging to the object, the user can mark its contour [CG98]. This is the principle of contour based extraction methods. The contour may either be defined as a set of control points or as a sketch of the

object. When only a few points are provided, they are automatically connected. The precision of the sketch is not critical. In the case of a rough sketch, an algorithm is required to adjust the boundaries to the real ones. These methods provide very precise object contours, but they are usually slower than feature-based techniques.

Region-based interaction — Image regions can finally be used to semi-automatically extract video objects. In this case, the user interacts with the result of a preliminary segmentation of the image to regions [RC98]. The user marks some of these regions as corresponding to a semantic object. These are then automatically merged to obtain the shape of the semantic video object.

User interaction provides a simple way of integrating semantics into the extraction process, and is more efficient than manual extraction, because it usually limits human intervention to one frame only. However, one of the disadvantages of a semi-automatic approach is the inability to detect new objects, as well as the fact that it is time consuming because of the necessity of user intervention.

2.2 Segmentation Strategy

A general scheme for segmentation can be seen as the concatenation of three major steps [PS99] represented in Fig.2.1: *Simplificat*ion, *Feature extraction* and *Decisio*n.

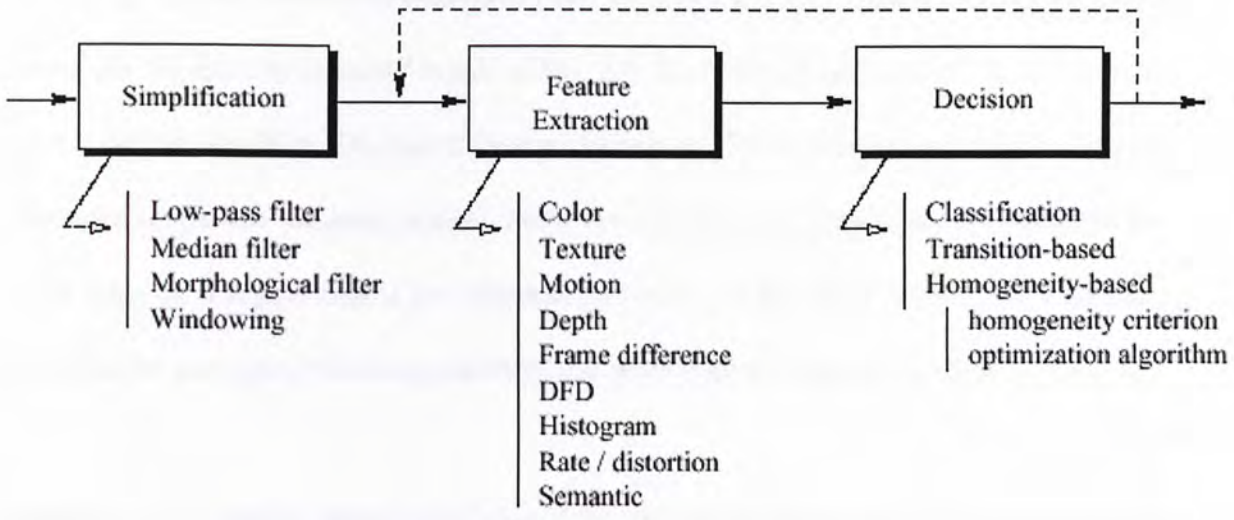


Figure 2.1 Major segmentation steps and related choices

Simplification — Most of the time, the original data in an image or in a video sequence contains information that is irrelevant for a given application. In such cases, data should be simplified by removing (e.g., filtering) irrelevant information. Furthermore, the simplified data should contain areas that are easier to segment. For instance, simplification can reduce the complexity of textured areas or remove details smaller than a given size. Typical filtering tools are listed below the *Simplification* block in Fig. 2.1. The simplification should not modify the boundary information that is relevant for the application.

Feature extraction — Segmentation is performed relying on specific features of the data. The selection of the *feature space* drives the type of homogeneity that is expected in the final partition. In some applications, the original data directly provides the feature space necessary for segmentation. For example, for color segmentation, the pixel values can directly correspond to the feature of interest. However, in a large number of cases, the

features of interest have to be estimated from the original data. Typical features are listed below the "Feature Extraction" block of Fig. 2.1. The list includes texture , motion, depth, Frame Difference (FD), Displaced Frame Difference (DFD), histograms or even spaces characterizing some semantic notions. Note that, in some cases, the feature estimation has to be done on a region that is homogeneous in terms of the same feature. As a result, a loop may be introduced to obtain the final result through an iterative process.

Decision — To finally obtain a partition of the data, the feature space has to be analyzed. The decision step decides on the position of the boundaries that form the partition in the decision space. Boundaries separate data areas that contain elements sharing the same characteristics in the selected feature space. For instance, in spatial segmentation, the decision may yield the precise shape of a region or, in temporal segmentation, the exact set of frames that form a shot.

A *region* created by a segmentation algorithm is defined as a set of elements (pixels or images) homogeneous in the feature space and connected in the decision space. A region may not have any semantic meaning. On the contrary, an *object* is the visual 2-D representation of an entity that has a semantic meaning. An object may be formed by the union of several regions.

Segmentation techniques often use more than one feature. This can be done either through the definition of a complex criterion combining several features or through the use of several segmentation steps that use different criteria. For example, the application

of various degrees of simplification allows the analysis to be done at several levels of resolution and, at each resolution level, a specific feature space may be used. Feature space can also be very complex to define if the segmentation process allows user interaction. In these cases, the user can implicitly introduce semantic notions that might not be easily obtained by any automatic analysis of the data. As a result, it is often not possible to classify the segmentation algorithms as a function of the feature space they use.

2.3 Segmentation of Moving Objects

A semantic video object is a collection of image pixels that correspond to the projection a real object in successive image planes of a video sequence. The meaning, i.e. the *semantics*, may change according to application. For example, in a building surveillance application, semantic video objects are people, whereas in a clothes shopping application, semantic video objects are the clothes of the person. Even this simple example shows that defining semantic video objects is a complex and sometimes delicate task.

Segmentation of moving objects in image sequence plays an important role in image sequence processing and analysis, for physical objects are often characterized by a coherent motion that is distinct from that of the background. Motion is a very useful feature for segmenting video sequence. Some motion segmentation algorithms are based on motion only, however it can also complement other features such as color, intensity,

or edges that are commonly used for segmentation of still images to achieve high boundary accuracy.

2.3.1 Motion

All the segmentation algorithms of moving objects are based on temporal changes in image intensities (more generally color). In fact, the observed 2D motions based on intensity changes may not be the same as the actual 2D motions. To be more precise, the velocity of observed or apparent 2D motion vectors is referred to as *optical flow* [AV89]. Optical flow can be caused not only by object motions, but also camera movements or illumination condition changes. Figure 2.2 illustrates two special cases. In the first example, a sphere with a uniform surface is rotating under a constant ambient light. Because every point on the sphere reflects the same color, the eye cannot observe any change in the color pattern of the imaged sphere and thus considers the sphere as being stationary. In the second example, the sphere is stationary, but is illuminated by a point light source that is rotating around the sphere. The motion of the light source causes the movement of the reflecting light spot on the sphere, which in turn can make the eye believe the sphere is rotating. The real motion is obviously zero because no 3-D motion is present, while the change in intensity induces optical flow and thereof apparent motion. The observed apparent 2D motion is referred to as optical flow in computer vision literature.

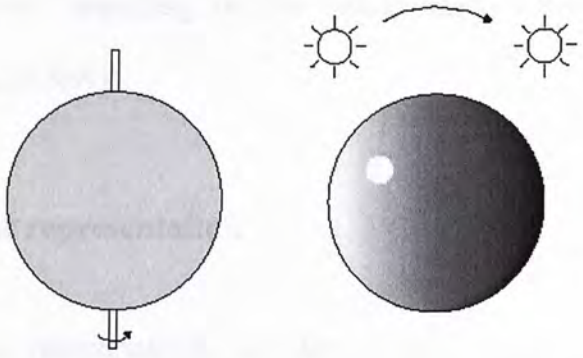


Figure 2.2 The optical flow is not always the same as the true motion field.

The above examples reveal that the optical flow may not be the same as the true 2D motion. However, when only image color information is available, the best one can hope to estimate accurately is the optical flow.

To segment a scene into independent moving objects, we need to know the real motion, but only apparent motion can be observed. As a result, it is normally more or less implicitly assumed that the real and apparent motions are the same, although it has been shown that they are in many cases different. Another important issue in motion estimation is noise sensitivity. It is easy to see that apparent motion is highly sensitive to noise, which can cause large discrepancies with respect to the real motion. However, we will use the term 2D motion or simply motion to describe optical flow, though sometimes it may be different from the true 2D motion.

2.3.2 Motion Field Representation

There are two ways of describing motion fields, which are non-parametric and parametric representation [KN99].

2.3.2.1 Non-parametric representation

In the nonparametric representation, we denote by $I(x, y; k)$ the intensity or luminance of pixel (x, y) in frame k. Apparent motion is what we perceive as motion and is induced by temporal changes in the intensity $I(x, y; k)$. It can be characterized by a correspondence vector field or by an optical flow field [HG90, AN79, DR84, JN91]. A correspondence vector describes the displacement of a pixel between two frames, whereas the optical flow (u, v) at the pixel $(x, y; k)$ refers to a velocity and is defined as

$$(u,v) = (\frac{dx}{dt}, \frac{dy}{dt}) \tag{2.1}$$

It is easy to see that optical flow and correspondence vectors are related.

In this presentation, the aperture problem (See Figure. 2.3) is tackled by incorporating a smoothness constraint that enforces neighbor pixels to have similar motion vectors. Block matching and its variants thereof are among the most popular non-parametric approaches due to their simplicity. In block-based motion estimation, the current frame is subdivided into blocks of equal size, and for each block the best match in the next (or previous) frame is computed. All pixels of a block are assumed to undergo the same translation, and are assigned the same correspondence vector. The selection of block size is depending on application. A large block size might contain more than one object with different motion directions and cannot accurately locate motion boundaries. In contrast,

small windows often result in wrong matches within uniform regions in the presence of noise.

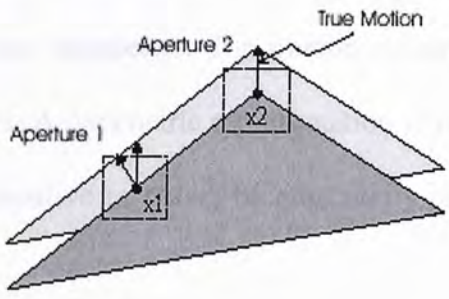


Figure 2.3 Illustration of the aperture problem. To estimate the motion at x1 using aperture 1, it is impossible to determine whether the motion is upward or perpendicular to the edge, because there is only one spatial gradient direction in this aperture. . On the other hand, the motion at x2 can be determined accurately, because the image has gradient in two different directions in aperture 2.

A drawback of non-parametric algorithm is the blurring of motion edges introduced by the smoothness constraint. This can pose a problem for segmentation techniques that are based solely on the estimated motion field. If the motion boundaries are blurred, then an exact boundary location cannot be expected. On the other hand, the rather generic assumption of smoothness makes non-parametric methods applicable for a broad range of situation and applications.

Non-parametric dense field representations are not directly suitable for segmentation because an object moving in the 3-D space generates a spatially varying 2-D motion field even within the same object, except for the simple case of pure translation. Hence, it would be difficult to group pixels based on the similarity of their flow vectors. That is why parametric models are commonly used in segmentation algorithms. However, dense field estimation is often the first step in calculating the model parameters.

2.3.2.2 Parametric representation

Parametric models require a segmentation of the scene and describe the motion of each region by a set of a few parameters. The motion vectors can then be synthesized from these model parameters. A parametric representation is more compact than a dense field description and less sensitive to noise, because many pixels are treated jointly to estimate a few parameters.

In order to derive a model or transformation that describes the motion of pixels between successive frames, assumptions on the scene and objects have to be made. Let (X,Y,Z) and (X',Y',Z') denote the 3-D coordinates of an object point in frame k and k+1, respectively. The corresponding image plane coordinates are (x, y) and (x', y'). If a 3-D object undergoes translation, rotation, and linear deformation, the 3-D displacement of a point on the object is given by [HG90]

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \underbrace{\begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix}}_{S} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \underbrace{\begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}}_{T} \tag{2.2}$$

T is a 3-D translation vector, while S is often defined as a 3×3 rotation matrix.

For motion estimation, real-motion objects are often approximated by piecewise planar 3-D surfaces. This, at least locally, is a reasonable assumption. The points on such a planar patch in frame k satisfy

$$aX + bY + cZ = 1 \tag{2.3}$$

If such a planar object is moving according to (2.2), the 6-parameter (affine) motion model is obtained under orthographic projection and the 8-parameter model under perspective projection.

❖ 6-parameter (affine) model

As can be seen from Figure 2.4, the 3-D coordinates are related to the image plane coordinates under the orthographic (parallel) projection by

$$(x, y) = (X, Y) \quad \text{and} \quad (x', y') = (X', Y') \tag{2.4}$$

This projection is computationally efficient and a good approximation if the distance between the objects and the camera is large compared to the depth of the objects. By combining (2.2)–(2.4), it follows that

$$x' = a_1 x + a_2 y + a_3 \tag{2.5a}$$

$$y' = a_4 x + a_5 y + a_6 \tag{2.5b}$$

with $a_1 = (s_{11} - s_{13}\frac{a}{c})$, $a_2 = (s_{12} - s_{13}\frac{b}{c})$, $a_3 = (t_1 + s_{13}\frac{1}{c})$, $a_4 = (s_{21} - s_{23}\frac{a}{c})$, $a_5 = (s_{22} - s_{23}\frac{b}{c})$, and $a_6 = (t_2 + s_{23}\frac{1}{c})$. Equation (2.5) is the well-known the affine motion model.

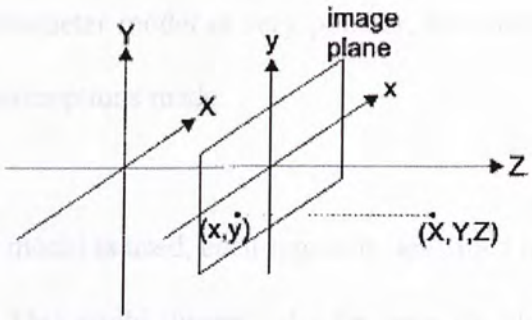


Figure 2.4 Projection of pixel (X, Y, Z) onto image (x, y) under orthographic (parallel) projection

❖ 8-parameter model

In the case of the more realistic perspective (central) projection, we can see from Fig. 2.5 that

$$(x, y) = (f\frac{X}{Z}, f\frac{Y}{Z}) \quad \text{and} \quad (x', y') = (f\frac{X'}{Z'}, f\frac{Y'}{Z'}) \tag{2.6}$$

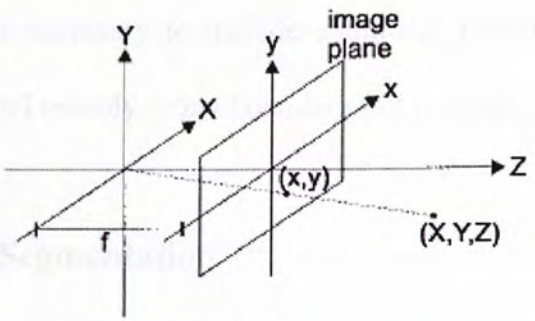


Figure 2.5 Projection of pixel (X, Y, Z) onto image (x, y) under perspective (central) projection

Together with (2.2) and (2.3), this results in the 8-parameter model

$$x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} \tag{2.7a}$$

$$y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1} \tag{2.7b}$$

Both the affine and 8-parameter model is very popular, but many other transformations exist depending on the assumptions made.

Independent of what model is used, each region is described by one set of parameters that must be estimated. This could theoretically be done by identifying corresponding point pairs in the two image frames. The 8-parameter model (2.7), for instance, require at least four independent point pairs to solve for the parameters. Unfortunately, to find such pairs automatically is a difficult task. As a result, the parameters are usually obtained either by fitting the model in the least-squares sense to a dense motion field obtained by a non-parametric method or directly from the luminance signal and gradient information. Although parametric representations are less noise sensitive, they still suffer from the intrinsic problems of motion estimation that depends on the exactness of estimated flow field. Most likely, it is necessary to include additional information such as color or intensity to accurately and reliably detect boundaries of moving objects.

2.3.3 Video Object Segmentation

The process of extracting the collections of image pixels corresponding to meaningful entities is referred to as video object segmentation. The main requirement of this process is *spatial accuracy*, that is, precise definition of the object boundary. The goal of the extraction process is to provide pixel-wise accuracy. Another basic requirement for semantic video object extraction is *temporal coherence*. Temporal coherence can be seen as the property of maintaining the spatial accuracy in time. The property allows us to

adapt the extraction to the temporal evolution of the projection of the object in successive images. Approaches were introduced for segmentation of video sequences into moving video objects that can be broadly classified into four categories: spatio-temporal, motion, morphological, and model-matching techniques. The different segmentation strategies will be reviewed and classified in the following.

2.3.3.1 Segmentation Based on Motion Information only

Many researchers have reported segmentation techniques that partition the scene based solely on motion information [MM94, MM97, CS97, ET00, BR02]. Traditional motion-based object segmentation methods, which employ motion information only, usually deal with scenes with rigid motion or piecewise rigid motion. It can be classified either based on their motion representations or based on their clustering criteria. Since motion representation plays such a crucial role in motion segmentation, motion-based segmentation techniques generally focus on the design of motion estimation. Therefore, motion segmentation is best identified and distinguished by the motion representation it adopts. The motion vectors encoded in MPEG-1 and 2 bitstreams are used to extract and track video object in [BR02]. Sparse motion field is found from accumulation of motions of marcoblocks from several frames. Two-dimensional median filter is applied to remove noise from the accumulated sparse motion vectors. Then the sparse motion field is interpolated using a surface interpolation method. It is assumed that the background is static, therefore, pixels with zero motion are assigned as background. The remaining pixels are clustered into different layers by applying Expectation-Maximization algorithm, with the number of object first estimated using K-means clustering algorithm.

A classical approach among these is the segmentation of an estimated dense motion field. But simply applying one of segmentation methods directly to the flow field does not produce useful result, because apart from the case of pure translation, a moving object generates a spatially varying flow field, like a walking person. So motion estimation and image segmentation should be jointly considered for better segmentation results.

2.3.3.2 Spatio-Temporal Segmentation

The spatio-temporal segmentation methods are comparatively new, which employ both spatial and temporal information embedded in the sequence to directly target the emerging multimedia applications and generic situations. By combining both motion and spatio-temporal information, these techniques intend to overcome the over-segmentation problem in image segmentation and the noise-sensitive problems in motion-based segmentation. The spatio-temporal segmentation is classified into motion segmentation because it employs the same motion estimation techniques as in motion-based segmentation and temporal segmentation is usually used to guide the overall segmentation results. However, this group of segmentation algorithms differs from the motion-based segmentation in that it makes use of spatial information to rectify and improve the temporal segmentation results. In this way, not only can it overcome the above-mentioned problems in motion-based segmentation but also can be adopted to non-rigid motion, therefore, more generic scenes. In the following, some typical segmentation methods will be presented.

Change detection [TM99, OE02, NH04, LQ03] is a temporal segmentation tool aiming at identifying changes in image sets or image sequences at two different times. Compared with motion estimation, it is relatively easy to compute. Different change detection techniques can be employed for moving camera and static camera conditions. If the camera moves, change detection aims at recognizing coherent and non-coherent moving areas. The former correspond to background areas, the latter to moving objects [TM99]. If the camera is static, the goal of change of change detection is to recognize moving objects and the static background [OE02]. [OE02] proposed an adaptive threshold estimation for CDM using the least-half-samples (LHS) technique. Change Detection based on F test is proposed in Nariman [NH04]. The threshold is calculated using the statistical information. Different from the previous statistical methods that rely on choosing thresholds, a new approach is presented in [LQ03], which does not require selection of empirical thresholds by modeling the change detection as an optimization problem by introducing MAP-MRF. Meier and Ngan [TM99] proposed a VOP segmentation scheme, which starts with motion estimation and compensation. The initial object model or initial object mask is obtained by combining the motion segmentation result from a complex *morphological motion filtering* or Change Detection Mask (CDM) with a spatial segmentation using *canny operator*. The object tracking is done by using *Hausdorff distance* and the model update process. The VOP extraction contains two steps. First, the closed VOP boundary is approximated by a simple filling-in technique. Then the wrong boundaries of VOP will be corrected.

Vasileios [VM04] proposed a spatio-temporal segmentation method based on initially applying an efficient two-dimensional (2-D) segmentation algorithm to the first frame of the image sequence, to produce an initial segmentation mask comprising regions which are homogeneous both in color and in motion. Following that, the formed regions are tracked in the subsequent frames. This allows for the temporal tracking to be performed at the homogeneous region level rather than the semantic object level, using the Bayes classifier for minimum mean-square error. Motion is handled in a different way, as it is utilized at the sequence level rather than the frame level for merging regions to semantic objects. In this way, generic objects can be easily tracked, by tracking their constituent homogeneous regions.

The proposed algorithm in [HX04] begins with the robust motion segmentation on the first two successive frames. To detect moving objects, segmented regions are grouped together according to their spatial similarity. Binary object models are derived from edge information. The binary object model for each moving object is automatically derived and tracked in subsequent frames using the generalized Hausdorff distance. The difficult task of using Hausdorff distance, to effectively distinguish between the background and object when updating the new model is also well contained within the technique.

Bayesian approach to video object segmentation is proposed in [YP05]. First, the video data are partitioned into a set of three-dimensional (3-D) watershed volumes, where each watershed volume is a series of corresponding two-dimensional (2-D) image regions. These 2-D image regions are obtained by applying to each image frame the marker-

controlled watershed segmentation. Next, a Markov random field is used to model the spatio-temporal relationship among the 3-D watershed volumes. Finally, the desired video objects can be extracted by merging watershed volumes having similar motion characteristics within a Bayesian framework.

A new fast watershed algorithm, named P-Watershed, for image sequence segmentation is proposed in [SY03]. The frame difference is thresholded to form an initial object mask, which detects the object position. The initial object mask is refined by eliminating the noise regions in the mask. Meanwhile, the predictive watershed process generates region label information; that is, the current frame is divided into many homogeneous regions, and each region is labeled with a unique label. Finally, regions are selected by the refined mask to form the object mask. If the percentage of pixels in the refined mask within a region is beyond a threshold, the whole part of the region is labeled as a foreground object; otherwise, it is regarded as a background region. In the shadow-cancellation mode, the gradient filter is used to suppress the effect of shadow and light changing. This segmentation algorithm is very similar with our proposed algorithm in next chapter. Both algorithms use spatial information to do region partition and temporal information to detect the position of moving objects. The results of it will be compared with ours in the chapter 3.

2.3.3.3 Morphological techniques

Morphological techniques for video object segmentation [PS94, MP94, DW98, JP97], which involve morphological filters or watershed segmentation techniques, are

computationally efficient and have gained increased popularity in recent years. These techniques typically start by a simplification step of the video frames using morphological filters. Then, a marker extraction step involves detecting the presence of homogeneous areas. Then, the undecided pixels are assigned a label in a decision step. Some of these algorithms consider luminance information only, e.g., [PS94], while others consider both spatial and temporal information [MP94, DW98, JP97].

2.3.3.4 Model-matching techniques

Starting with an initial object model, model-matching segmentation techniques aim to locate the object in the video scene based on the best match between a model of the object and the scene's frames. In general, a robust model-matching approach should address the issues of object occlusion, object deformation, and multiple moving objects in the presence of noise. An approach utilizes a multiple feature template that includes color, texture, edge, and motion information [YZ00]. Other approaches rely on minimizing the interframe difference density function, approximated using a statistical model [NP00], or the generalized Hausdorff distance for determining the resemblance of one point set to another [DP93]. An object tracking technique was developed where the model of the video object, known a prior, was matched against subsequent frames by minimizing the Hausdorff distance [DP92]. Another technique based on the Hausdorff distance involves extracting an initial object model in the dense optical flow [TM98]. Then, the model is updated using the concept of moving connected components, which are assumed to represent a single moving object against a stationary background.

2.3.3.5 Specific Applications of Video object Segmentation

2.3.3.5.1 Face Segmentation

The task of finding a person's face in a picture seems effortless for humans to perform. However it is far from simple for machine of current technology to do the same. Many existing methods only work well on simple images with benign background and frontal view of the person's face. To cope with more complicated images and conditions, many more assumptions will have to be made. In the literature, large number of face segmentation algorithms based on different assumptions and applications have been reported. According to the primary criterion for segmentation, two categories can be classified: color-based methods [DC99, NH04] and facial features-based methods [HL03].

Color information has been introduced to the face segmentation problem because the human skin has a color distribution that differs significantly, although not entirely, from those of the background objects. Chai [DC99] introduced a universal skin-color map which uses chrominance component to detect pixels with skin-color appearance. In order to overcome the limitations of color segmentation, some processing stages are employed to refine the output result. Nariman [NH04] presented a hand and face segmentation methodology using color and motion cues for the content-based representation of sign language video sequences. The skin-color distribution in the CbCr plane is modeled as a bivariate normal distribution. Image pixels are classified as skin if the Mahalanobis distance between their feature vectors and the mean vector of the skin class is less than a predetermined segmentation threshold, which is derived by minimizing the probability of

error. The aim of change detection is to localize moving objects in a video sequences, which will be combined with the results from skin-color segmentation to extract face and hands.

The facial features-based method, on the other hand, utilizes the facial statistical or other structural features rather than skin-color information to achieve face segmentation. An independent component analysis based approach [SZ05] is presented for learning view-specific subspace representations of the face object from multiview face examples. This method takes into account higher order statistics needed for object view characterization. In addition, a method [AN05] within the framework of principal component analysis is also proposed to recognize faces in the presence of clutters. In order to improve performance in the presence of background, the authors argue in favor of learning the distribution of background patterns and show how this can be done for a given test image. A statistical model-based video segmentation algorithm is presented for head-and-shoulder type video in [HL03]. This algorithm uses domain knowledge by abstracting the head-and-shoulder object with a blob-based statistical region model and a shape model. The object segmentation problem is then converted into a model detection and tracking problem.

2.3.3.5.2 Depth segmentation

Stereo–video archives are anticipated to rapidly increase in the forthcoming years. However, traditionally stereo–image sequences are represented by numerous consecutive image frame pairs), each of which corresponds to a certain time instant. In [MA00], a

novel framework for object extraction is described from images utilizing multiple cameras. Focused regions in images and disparities of point correspondences among multiple images are 3-D clues for the extraction. Edges in images captured by the cameras are detected, which is a key to identify the focused object and disparity of the edges are used to extract depth information. A focused object is extracted from an image as a set of edge intervals with the disparity keys. Nikolaos [ND00] proposed an algorithm of extracting objects using depth information. Depth is estimated from the disparity field between the left and right channel images. A multiresolution implementation of the RSST algorithm (M-RSST) is presented to perform both color and depth segmentation. The color segments are projected onto depth segments so that video objects identified by depth segmentation are retained, while at the same time accurate object boundaries are extracted.

Low DOF is an important technique widely used by professional photographers for various types of images, such as telephoto images, to emphasize a certain object. It is also a key technique for microbiologists to understand the 3D structure within a specimen under a high-power microscope. Some work focus on the segmentation of low depth of field (DOF) image. A novel algorithm to partition an image with low DOF into focused object-of-interest (OOI) and defocused background. James [JZ01] proposed a multi-scale approach based on high frequency wavelet coefficients and their statistics is used to perform context-dependent classification of individual blocks. Changick [CK05] transformed low-DOF into an appropriate feature space, in which the spatial distribution of the high-frequency components is represented. This is conducted by computing higher

order statistics (HOS) for all pixels in the low-DOF image. Then, the obtained feature space, which is called HOS map, is simplified by removing small dark holes and bright patches using a morphological filter by reconstruction. Finally, the object of interest is extracted by applying region merging to the simplified image and by thresholding.

2.4 Summary

Digital video is an integral part of many newly emerging multimedia applications. New image and video standards, such as MPEG-4 and MPEG-7, not only concentrate on efficient compression methods but also on providing better ways to represent, integrate, and exchange visual information. These efforts aim to provide the user with greater flexibility for "content-based" access and manipulation of multimedia data. In order to obtain a content-based representation, an input video sequence must first be segmented into an appropriate set of arbitrarily shaped objects, termed the video object planes (VOPs) in the MPEG-4 Verification Model, with each object possibly representing a particular meaningful content of the video stream.

Many approaches have been proposed for video segmentation, both semi-automatic and automatic. The former requires human interaction for defining the number of objects present in the sequence or more often for grouping homogeneous regions to semantic objects, while the latter requires no such interaction. But fully automatic video object segmentation is very difficult, because objects are not homogeneous with respect to low-level features but involve higher-level semantic concepts. Motion is important cue to do

video object segmentation, but non-rigid objects usually do not exhibit coherent motion, so motion information should be combined with other features to extract objects.

Chapter 3
Automatic Video Object Segmentation Algorithm

This proposed unsupervised video segmentation algorithm is based on the *k*-medians clustering method [SA98] and object tracking. It uses motion and color information to extract moving objects, which can tackle the problem of using only the region or boundary information. The major advantage of the *k*-medians clustering method is that it is less sensitive to outliers as compared with the *k*-means clustering method. In our proposed method, clustering is first employed to segment the frame of a video sequence into homogeneous regions based on luminance, chrominance, texture and motion information. In order to deal with the over-segmentation problem, region merging is employed. Then, two methods of object tracking can be used. The first one is to set up a binary model of moving object. A binary model is derived for the object of interest and tracked throughout the sequence. Temporal continuity of the segmentation is accomplished by matching the model against subsequent frames and updating them accordingly. This allows the proposed method to track fast moving objects as well as to detect the objects that stop moving in the scene. The other method is to refine the regions by using a combination of features according to a set of appropriate rules. Regions belonging to moving objects are tracked by using region descriptors from the current frame to the next frame. This algorithm is capable of dealing with multiple simultaneous objects. Defining the tracking based on the parts of objects, identified by region

segmentation, has led to a flexible technique that exploits the nature of the video object tracking problem.

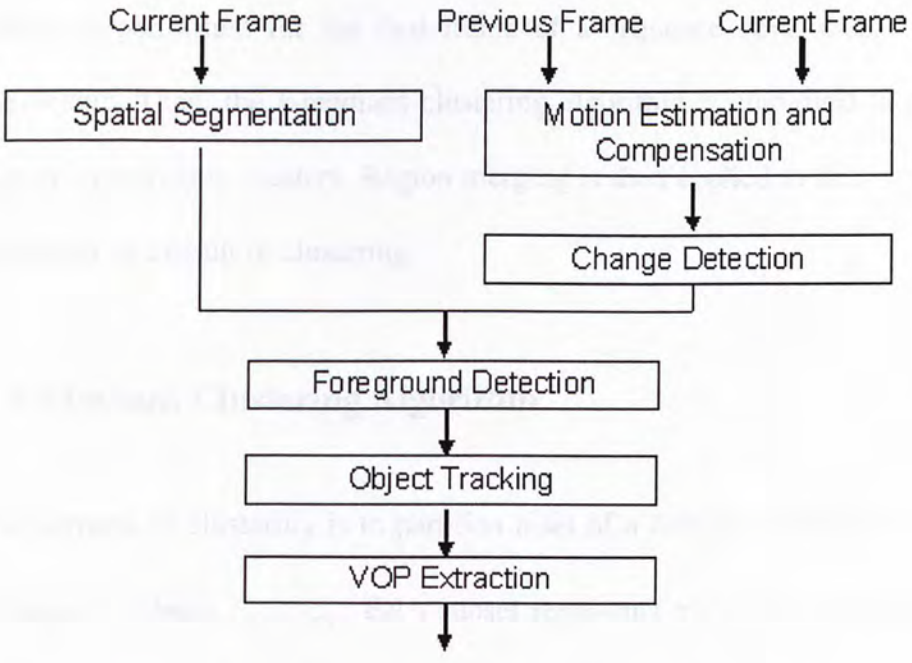


Figure 3.1 Block diagram of our VOP segmentation algorithm

This chapter is organized as follows. In Section 3.1, we present spatial segmentation based on the weighted *k*-medians clustering algorithm and region merging. Section 3.2 introduces the method of tracking and extracting moving objects. Our method is evaluated on real sequences in Section 3.3, and the algorithm is concluded in Section 3.4.

3.1 Spatial Segmentation

The goal of spatial segmentation is to partition an image into a set of disjoint homogeneous regions whose union is the entire image. First, the feature vectors are

extracted for each pixel in frame $F(x, y, t)$, and is presented in an appropriate form. x and y represent the spatial location of a pixel, and t the frame number. Second, the number of clusters inherent in the image is estimated. Note that the cluster number estimation is performed for the first frame of a sequence only, unless a scene change occurs. Then, the k-medians clustering algorithm is employed to partition the feature vectors into clusters. Region merging is then applied to deal with over-segmentation as a result of clustering.

3.1.1 k-Medians Clustering Algorithm

The purpose of clustering is to partition a set of n feature vectors $C = \mathbf{u}_1,...,\mathbf{u}_n$ into k disjoint subsets, $C_1,...,C_k$. Each subset represents a cluster, with the feature vectors in the same cluster being more similar to each other than to the feature vectors in other clusters. Generally, C is partitioned by optimizing some criterion function. The most popular criterion function for clustering is the sum-of-squared-error criterion. Let m_i be the median of those samples,

$$\mathbf{m}_i = \underset{\mathbf{u}\in C_i}{median}(\mathbf{u}) \tag{3.1}$$

Let us consider an N_f - dimensional feature space. The sum-of-squared errors is defined by

$$J_k = \sum_{i=1}^{k} \sum_{\mathbf{u}\in C_i} d^2(\mathbf{u}, \mathbf{m}_i), \tag{3.2}$$

where

$$d(\mathbf{u},\mathbf{m}_i)=\|\mathbf{u}-\mathbf{m}_i\|=\left[\sum_{l=1}^{N_f}(u_l-m_{il})^2\right]^{\frac{1}{2}} \tag{3.3}$$

J_k measures the total squared error incurred in representing the n samples $\mathbf{u}_1,...,\mathbf{u}_n$ by the k cluster centers, $\mathbf{m}_1,...,\mathbf{m}_k$. The value of J_k depends on how well the samples are grouped into clusters and the number of clusters. The optimal partitioning is defined as one that minimizes J_k.

The k-medians algorithm, given by the following steps, aims to minimize (3.2).

1. Select k initial cluster centers $\mathbf{m}_1,...,\mathbf{m}_k$.
2. At the pth iterative step, assign each feature vector to one of the k clusters according to the relation

$$\mathbf{u}\in C_i^{(p)} \text{ if } \left\|\mathbf{u}-\mathbf{m}_i^{(p)}\right\|<\left\|\mathbf{u}-\mathbf{m}_j^{(p)}\right\| \tag{3.4}$$

for all $j=1,...,k, j\neq i$. That is, assign each sample to the class of the nearest cluster center.

3. Update the cluster center $\mathbf{m}_i^{(p+1)}$, $i=1,...,k$, as the sample median of all samples in $C_i^{(p)}$

$$\mathbf{m}_i^{(p+1)}=\underset{\mathbf{u}\in C_i^{(p)}}{median}(\mathbf{u}) \tag{3.5}$$

4. If $\mathbf{m}_i^{(p+1)}=\mathbf{m}_i^{(p)}$ for $i=1,...,k$, the algorithm has converged and the procedure is terminated. Otherwise go to step 2.

The k-medians algorithm converges to the local minimum of J_k.

3.1.2 Cluster Number Estimation

A fundamental problem with the *k*-medians algorithm is the lack of knowledge to identify the number of clusters present in the data. A clustering algorithm will generally converge to any given number of clusters, even if there are no actual clusters present in the data. Therefore, a criterion, or a cluster quality measurement, has to be defined so that the cluster number that optimizes this measurement could be assumed to reflect the number of clusters inherent in the data.

In our study, the cluster number is estimated by analyzing the behavior of J_k for $k = 1, \ldots, k = k_{\max}$ for the first frame. It is clear that J_k, which measures the total squared error incurred in representing the n samples $\mathbf{u}_1,...,\mathbf{u}_n$ by the k cluster centers $\mathbf{m}_1,...,\mathbf{m}_k$, must decrease monotonically as k increases, because the squared error can be decreased each time k is increased merely by transferring a single sample to new singleton cluster. If the n samples are really grouped into $\hat{k}$ compact, well-separated clusters, J_k should decrease rapidly until $k = \hat{k}$, and then decrease much more slowly thereafter until it reaches zero at $k = n$. Based on this property, the true number of clusters must lie at the "corner" or "elbow" of the J_k versus k curve [YH94] [NN03].

3.1.2.1 Initial Cluster Number Estimation

First, only luminance information is used to obtain an estimation of the cluster numbers in the first frame. The *k*-medians algorithm is running for a range of

different cluster number $k = 1, 2, 3 \ldots, k = k_{max}$ and J_k evaluated after convergence. The starting points for the $(k+1)$th cluster were derived from the centers of the kth cluster, plus the sample that is farthest from the nearest cluster center. The first k that satisfies the following conditions is designated as the cluster number:

$$\frac{\Delta J_k}{\Delta J_{k+1}} < \theta \tag{3.6}$$

where,

$$\Delta J_k = J_k - J_{k+i} \tag{3.7}$$

and θ is some predefined threshold. From the J_k versus k curve in Figure 3.2(a), we can see that when k is larger than the "corner" or "elbow" number, the J_k changes slowly, and then θ should almost be 1, it is set to 1 in our implementation. The value of i must be selected to ensure that the cluster number is selected as the true cluster number which would lie at the corner of the J_k versus k curve, rather than at a *mini-corner*, so it is set to 2. Figure 3.2 (a) shows the J_k versus k curve for the first frame of the *Mother & Daughter* sequence. Figure 3.3(c) depicts the corresponding segmentation field.

3.1.2.2 Cluster Number Estimation with Multiple Features

There are two problems to be solved before clustering the image. Firstly, the features have quite different ranges of possible values. Thus, normalization of different features should be carried out. Secondly, the features used in the scheme

differ not only in their values but also in the level of reliability. It means that different features should be assigned different weights.

Feature Extraction

The features pertaining to each pixel are represented in the form of a feature vector. The features that we have considered are luminance (Y), chrominance (C_b and C_r), texture and motion. The luminance and chrominance features are derived directly from the input image. In order to alleviate the effect of impulse noise, the Y, C_b, and C_r components are pre-processed with a median filter [http1] of size 5×5 pixels.

The texture is a very important feature. In our algorithm, it is derived using J-image [YD01], whose advantage is to consider the space distribution of pixels in a small window.

Motion information is derived using the diamond search block matching algorithm [JY98]. The motion estimation results of the diamond search algorithm are comparable to that of the full search algorithm, however the diamond search algorithm is computationally less expensive. A median filter is employed to remove the outliers in the video frame, which ensures that motion detection will not be affected by the outliers in the gray-scale data.

The normalization of different features

The features that we propose to use are characterized by quite different ranges of possible values: luminance and chrominance values typically range from 0 to 255, the texture information shows the biggest variations, while motion spans a more limited interval (for example, [-10···+10] pixels/frame). In order to process this information in parallel, it is therefore necessary to introduce some form of normalization that allows us to define a distance that is easily measurable. A common solution is to normalize with respect to the standard deviation over the entire image, which is adopted in [RC98, BH02, PS96]. For each feature f_l,

$$m_l = \frac{1}{N}\sum_{q=1}^{N} x_{ql} \tag{3.8}$$

$$\sigma_l^2 = \frac{1}{N}\sum_{q=1}^{N} (x_{ql} - m_l)^2 \tag{3.9}$$

$$\hat{f}_l = \frac{f_l}{\sigma_l} \tag{3.10}$$

for all $l = 1,...,k$, where N is the total pixels in the whole frame and x is the pixel in the current frame.

The weight of different features

The method employed to adaptively assign weights to different features is important to the segmentation scheme. At first *k*-medians clustering algorithm based on luminance information is applied to obtain the initial regions. The variance

of the medians of the corresponding feature in the region $R_m^{(1)}$, $m = 1,...,M$, will affect the weights assigned to luminance, chrominance and texture information. The motivation for the definition of weights comes from the reasoning that more significant feature contributes more in representing the difference among the regions, so it is preferable to assign higher weight to more significant feature. For this reason, the weight of each feature should be proportional to the variance of the corresponding feature. At the same time, because different features have different ranges, the variances' ranges will be different. We define the weight of each feature as the normalized variance of the corresponding feature, as shown in (3.14).

$$\hat{f}_{ml} = \underset{\hat{f}_l \in R_m^{(1)}}{median}(\hat{f}_l) \tag{3.11}$$

$$\bar{f}_{ml} = \frac{1}{M}\sum_{m=1}^{M}(\hat{f}_{ml}) \tag{3.12}$$

$$\sigma_{ml}^2 = \frac{1}{M}\sum_{m=1}^{M}(\hat{f}_l - \bar{f}_{ml})^2 \tag{3.13}$$

$$w_l = \frac{\sigma_{ml}}{\text{range}(f_l)} \tag{3.14}$$

$$\text{range}(f_l) = \max_N(f_l) - \min_N(f_l) \tag{3.15}$$

where M is the total number of regions in the first frame and N is the total pixels in the whole frame.

The inclusion of the standardization and weighting terms modifies (3.3) to:

$$d'(\mathbf{u},\mathbf{m}_i)=\left[\sum_{l=1}^{N_f}(w_l\times(u_l-m_{il}))^2\right]^{\frac{1}{2}} \tag{3.16}$$

The k value will be obtained based on the multiple features, which is the same way as how the initial cluster number is estimated in section 3.1.2.1. The weight of motion information is computed in the similar fashion as the luminance, chrominance and texture information. Because in every frame, the motion information is quite different, the weight of motion is computed adaptively by using the region information obtained in the previous frame. But the weights of other features are decided in the first frame, unless a scene change occurs. If a scene change is detected, all the parameters will be reset. In this work, we assume that the video data has been parsed into shots. Within each shot, the video scene is continuous and there are no abrupt changes. In the implementation, the method described in [WJ01] can be used for video shot detection. Figure 3.2 (b) shows the J_k versus k curve for the first frame of *Mother & Daughter* sequence based on multiple features. Figure 3.3 (d) depicts the corresponding segmentation fields.

3.1.2 Region Merging

As mentioned in the previous section, the clustering algorithm is applied to partition an image into small regions that are homogeneous in terms of the multiple features. This process may result in over-segmentation, as shown in Figure 3.3(e).

To solve the over-segmentation problem, the region merging method will group sub-regions into larger regions based on a similarity criterion.

For every two neighboring regions R_i and R_j, we compute σ_i and σ_j, the standard deviations of R_i and R_j, respectively. We further calculate the standard deviation of the union of R_i and R_j as defined below:

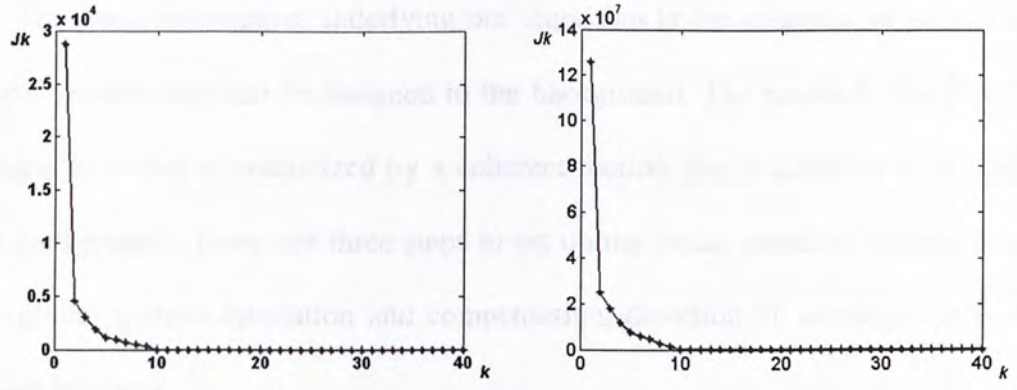
$$\mathbf{m}_{i,j} = \frac{1}{(n_i + n_j)} \sum_{u \in (R_i \cup R_j)} \mathbf{u} \tag{3.17}$$

$$\boldsymbol{\sigma}_{i,j}^2 = \frac{1}{(n_i + n_j)} \sum_{u \in (R_i \cup R_j)} (\mathbf{u} - \mathbf{m}_{i,j})^2 \tag{3.18}$$

where n_i and n_j are the number of pixels in regions R_i and R_j, respectively. The decision rule of region merging is then given by

$$\sigma_{(i,j)l} \leq \min(\sigma_{il}, \sigma_{jl}) \tag{3.19}$$

for all $l = 1, \ldots, k$. That is, if the variance $\sigma_{(i,j)l}$ of the combined regions is smaller than the minimum of the two regions R_i and R_j, these two regions will be merged. Figure 3.3(e) shows the region merging results of Figure 3.3 (d).



(a) (b)

Figure 3.2 Frame 1 of the *Mother & Daughter* sequence: (a) The *Jk* versus *k* curve based on color quantization map, (b) The *Jk* versus *k* curve based on multiple weighted features.

(a) (b) (c)

(d) (e)

Figure 3.3 Frame 1 of the *Mother & Daughter* sequence: (a) Original frame. (b) Result of color quantization. (c) Segmentation fields using luminance information. (d) Segmentation fields with multiple feature. (e) Region merging result.

3.2 Foreground Detection

The main assumption underlying our algorithm is the existence of a dominant global motion that can be assigned to the background. The reason is that physical objects are often characterized by a coherent motion that is different from that of the background. There are three steps to set up the initial model of objects, which are global motion estimation and compensation, detection of moving objects and object tracking.

3.2.1 Global Motion Estimation

In most cases, the global motion is very simple and consists only of pan and possibly zoom. Therefore, the six-parameter affine transformation [YW01]

$$x' = a_1 x + a_2 y + a_3 \tag{3.20a}$$

$$y' = a_4 x + a_5 y + a_6 \tag{3.20b}$$

is normally sufficient to describe the global motion. In this step, after the diamond search block-matching process has been performed, the least square (LS) method [TH86] is used to estimate the six parameters by minimizing the sum of the squared residuals. Because a pixel may not experience only global motion, the lack of robustness against outliers is a major drawback of this method. Usually, the motion of a pixel can be decomposed into a global motion and a local motion due to the global and local motion of the objects. Therefore, the prediction error obtained by using the global motion model alone may not be small, even if the correct global motion parameters are available. In other instances, not all the pixels in the same frame experience the global motion and ideally one should not apply the same

motion model to the entire frame. These problems can be overcome by a *robust estimation* method [PJ87], if the global motion is dominant over other local motions, in the sense that the pixels that experience the same global motion occupy a significantly larger portion of the underlying image than those pixels that do not.

The basic idea of how to obtain the six parameters is to consider the pixels that are governed by the global motion as *inliers*, and the remaining pixels as *outliers*. Initially, we assume that all the pixels undergo the same global motion, and estimate the motion parameters by the LS method over all of the pixels. This will yield an initial set of motion parameters. With this initial solution, the prediction or fitting error over each pixel can be calculated. The pixels where the errors exceed a certain threshold will be classified as *outliers* and be eliminated from the next iteration. The above process is then repeated for the remaining *inlier* pixels. This process iterates until no *outlier* pixels exist, that is, the estimates of six parameters converge.

3.2.2 Detection of Moving Objects

After compensating the background motion, the areas that do not follow this background motion indicate the presence of independently moving physical objects, which are named *Independently Moving Components* (IMC). Figure 3.6 (b) shows the area corresponding to the moving objects.

3.3 Object Tracking and Extracting

Temporal continuity and linking are two crucial aspects of VOP segmentation. The former requires the shape of the extracted VOPs to be a smooth function of time, whereas the latter ensures that objects do not get lost even when they stop moving, and it also enables applications to identify objects in subsequent frames. Two methods of object tracking are proposed in our algorithm: Binary Model Tracking and Region Descriptor Tracking.

3.3.1 Binary Model Tracking

A binary model is derived for the object of interest and tracked throughout the sequence. Temporal continuity of the segmentation is accomplished by matching the model against subsequent frames and updating them accordingly. This allows the proposed method to track fast moving objects as well as to detect the objects that stop moving from the scene.

3.3.1.1 Model Initialization

Initially, the position of the object is unknown and no models exist. With the assumption that physical objects are characterized by a different motion from that of the background, the IMCs obtained by the previous stage 3.2.2 can be utilized for model initialization. However, it is necessary to get sufficient evidence for an independently moving object to increase robustness and to avoid tracking noise. This is accomplished by requiring the IMCs to have a specified minimum size [TM99]. Edge is a very important cue to derive the exact object model for its

robustness to noise. The Canny operator [JC86] compares favorably with other edge detection methods. Thus, we will use it to obtain the initial model.

The Canny Operator detects the edges of a gray scale image. But in regions where the intensity variations are small, adjacent regions are so ambiguous that the Canny Operator will fail to detect the edges well. To overcome this problem, we will incorporate chrominance information in the edge detection process to obtain more accurate edge maps. As a result, the boundaries of the independently moving objects will be detected more accurately in the following steps. Chrominance information is incorporated into the gradient computation in a way that the largest values among the gradients are obtained. The input image processed by the Canny Operator is modified as follows:

$$Y' = \frac{1}{w_Y + w_{C_b} + w_{C_r}} (w_Y Y + w_{C_b} C_b + w_{C_r} C_r) \tag{3.21}$$

where w_Y, w_{C_b}, and w_{C_r} are weighting factors applied to Y, C_b and C_r, respectively, which are obtained in (3.14). The effectiveness of this modification is shown in Figure 3.4.

(a) (b)

(c) (d)

Figure 3.4 Frame 1 of the *Mother & Daughter* sequence: (a) Original frame. (b) Edge map based on luminance. (c) The input image processed by the Canny Operator. (d) Edge map based on multiple weighted features.

3.3.1.2 Initial Model Extraction

Suppose the model for the object of interest is initialized in frame i. Initial model $O_i = \{o_1, ..., o_m\}$ is defined as a set of m model points. Similarly, $E_i = \{e_1, ..., e_n\}$ is denoted as the set of all edge pixels detected by the Canny operator in frame i. Once E_i is computed, O_i can be derived. Since the correct object boundaries are in the occluded zones, or are at least in their vicinity, we are able to detect the boundaries of independently moving objects by selecting edge pixels near the IMCs. If the IMC denotes the set of all pixels belonging to this independently moving component, the initial model is given by selecting all edge pixels within a small distance T_{init} of IMC, i.e.,

$$O_i = \{e \in E_i \mid \min_{x \in IMC} \|e - x\| \leq T_{init}\} \tag{3.22}$$

where x is an element of IMC and $\|\cdot\|$ is the Euclidean distance. Equation (3.22) can be efficiently implemented using a distance transformation [GB86]. The resulting

model initialization for the *Mother & Daughter* sequence is shown in Figure 3.6(c). Note that only edges belonging to moving objects were detected as moving. In our implementation, T_{init} is set to 1 pixel.

Furthermore, only those components that are undergoing some motion relative to the background can be detected. Parts of an object that do not move relative to the background will be assigned to the background. Accordingly, it might take several frames to identify the whole object. This is achieved by the model update and can be considered as temporal integration.

3.3.1.3 Model Update

The object of interest might rotate or change its shape as it is moving through the video sequence, and as a consequence the corresponding model must be updated every frame. More precisely, the model is actually not updated, but a new model is derived by selecting an appropriate set of edge pixels from the edge image of the current frame. However, the object model of the previous frame is an important cue for choosing the set of edge pixels forming the new model.

The updated model is given by combining the two components, which are the *Updated Existing Component* and the *Newly Appearing Component*. The purpose of *Updated Existing Component* is to track the binary model of the previous frame using motion compensation and *Newly Appearing Component* is employed to incorporate newly appearing object.

Updated Existing Component

The *Updated Existing Component* case considers the quasi-rigid parts of an object that exhibit only minor changes in successive frames. Since these parts are not expected to change significantly by definition, they can be updated based on the old model O_q of frame q. If e_q belongs to the object binary model O_q of the previous frame, note that given a motion vector v at e_q, e'_{q+1} is obtained by $e'_{q+1} = e_q + v$.

$$O^E_{q+1} = \{x \in E_{q+1} \mid x \in \eta_{e'_{q+1}}\} \tag{3.23}$$

with $\eta_{e'_{q+1}}$ being a neighborhood centered on e'_{q+1}.

Newly Appearing Component

Newly Appearing Components are to incorporate non-rigid moving and newly appearing parts of an object into the model update. It is obtained by identifying IMCs in the same way as for the model initialization. All edge pixels in IMCs are added to the new model O_{q+1}. Formally, the *Newly Appearing Component* O^N_{q+1} is defined as

$$O^N_{q+1} = \{e_{q+1} \in E_{q+1} \mid e_{q+1} \in IMC\} \tag{3.24}$$

Lastly, the updated model O_{q+1} is given by combining the two components

$$O_{q+1} = O^E_{q+1} \cup O^N_{q+1} \tag{3.25}$$

Depending on the motion characteristics of the sequence and the objects, which can change from frame to frame, either O^E_{q+1} or O^N_{q+1} can dominate the update process.

This operation is a kind of temporal integration of the model. The component O_{q+1}^{N} detects and includes newly appearing parts of the objects, whereas O_{q+1}^{E} is the "memory" that stores the accumulated model components. Figure 3.6 shows that this technique yields a robust mechanism that can handle both objects that change quickly and objects that stop moving. The independently moving components in Figure 3.6(e) show that some parts of the objects stop moving in the current frame 23. However, they will persist when updating the model by combining the quasi-rigid and non-rigid parts, which is shown in Figure 3.6(f).

3.3.1.4. VOP Extraction

The output of the model update stage is a sequence of binary edge images that model the tracked objects. The remaining step is to extract the corresponding objects from the video sequence. Unfortunately, VOP extraction is not straightforward because the binary model points normally do not form a closed contour.

A two-step process determines the shape of the objects. Firstly, the initial VOPs are obtained using a simple filling-in technique. Secondly, the homogenous regions obtained in section 3.1 are matched with the initial VOPs to extract the moving objects from the sequence.

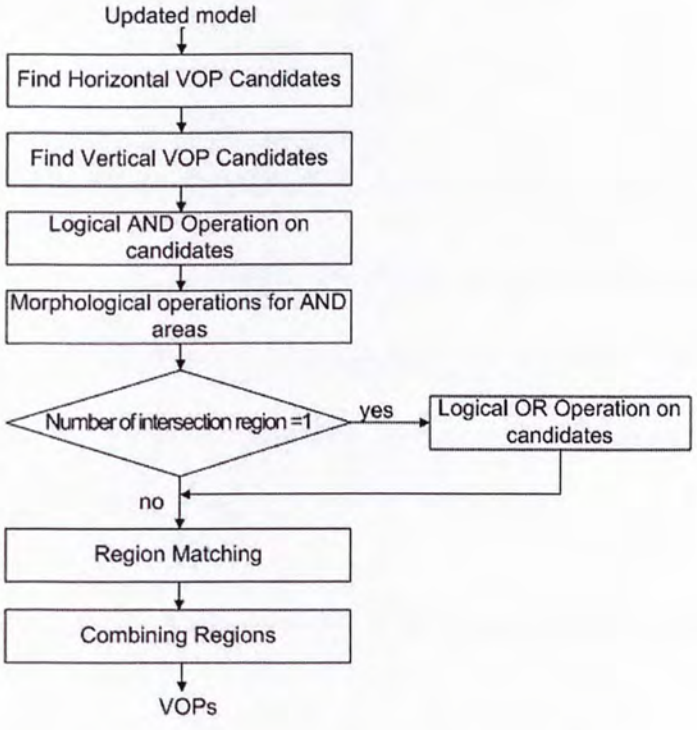


Figure 3.5 Flow chart of VOPs Extraction

A simple filling-in technique is employed to determine the initial VOPs. The horizontal candidates are declared to be the region inside the first and last edge points in each row and the vertical candidates for each column. After finding both horizontal and vertical candidates, the intersection regions through the logical AND operation are further processed by alternative use of morphological operators (the morphological close and open filters are used to smooth object boundaries and eliminate small regions).

Let R_i^t be a homogenous region. VOP V_j^t denotes the filled VOP in the current frame t. If R_i^t belongs to moving objects, it should share partially the same spatial locations with the filled VOP V_j^t. A decision rule of region matching is defined as

$$T_m = \frac{N_{R_i^t \cap V_j^t}}{N_{R_i^t}} \tag{3.26}$$

where $N_{R_i^t}$ is the number of pixels in R_i^t. Usually, when a large proportion of a region belonging to a moving object, this region should be a part of this moving object. So, if $T_m > T_r$, the region R_i^t is considered to belong to the VOP; otherwise it is the background. The value of T_r directly influences the segmentation results. In our implementation, T_r is set to 0.6.

After combining all the regions that belong to moving objects, then VOPs will be extracted.

(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.6 The *Mother & Daughter* sequence: (a) Original frame1. (b) Independently moving component of frame1. (c) Initial model. (d) Original frame 23. (e) Independently moving component for frame 23. (f) Updated binary model for frame 23. (g) Initial VOPs after morphological close and open operators (h) Segmentation fields. (i) Result of region matching

3.3.2 Region Descriptor Tracking

Video object tracking algorithms should be able to deal with the various dynamics in the scene. The goal is to establish a stable track for each object. Using the region descriptor is another proposed method to track objects. Projecting a region descriptor instead of the entire region is a simple and effective strategy. The simplicity comes from the fact that instead of projecting the entire region into the next frame, only the region descriptor needs to be processed. Therefore, there is no need for computationally expensive motion models for moving objects. In addition, region descriptor projection is effective, since it can cope with deformation and complex motion, when updating the feature values in the region descriptor by refining the predicted region partition.

3.3.2.1 Region Descriptors

A region defines the topology of pixels that are homogeneous according to a specific criterion. The homogeneity criterion is defined with respect to one or more features in the dense feature space. The values of the features characterizing the region are distinctive of the region itself. We summarize these feature values in a vector, henceforth referred to as region descriptor. Region descriptors are the simplest way of representing the characteristics of region. A region descriptor $\boldsymbol{\varphi}_r^{(t-1)}$ computed for every moving region $R_r^{(t-1)}$ in frame t-1, is represented as

$$\boldsymbol{\varphi}_r^{(t-1)} = \underset{x \in R_r^{(t-1)}}{median}(\mathbf{u}) \tag{3.27}$$

Note that the region descriptor is a vector where each element of the vector is the median of the corresponding feature in the region $R_r^{(t-1)}$

$$\boldsymbol{\varphi}_r{}^{(t-1)} = (\varphi_{r,1}^{(t-1)}, \varphi_{r,2}^{(t-1)}, \varphi_{r,3}^{(t-1)}, \varphi_{r,4}^{(t-1)}, ..., \varphi_{r,l}^{(t-1)},)^T \tag{3.28}$$

where l is the number of features in the frame . Let $\varphi_{r,1}^{(t-1)}$, $\varphi_{r,2}^{(t-1)}$ represent the position of the region descriptor, and $\varphi_{r,3}^{(t-1)}$, $\varphi_{r,4}^{(t-1)}$ be its motion vector. The position and the motion vector of the region descriptor are given by the median values of the pixels and their associated motion vectors belonging to the corresponding region.

In our implementation, l=8. In particular, $\varphi_{r,5}^{(t-1)}$, $\varphi_{r,6}^{(t-1)}$, $\varphi_{r,7}^{(t-1)}$ represents the median value of *Y*, *Cb* and *Cr* components in the corresponding region, and $\varphi_{r,8}^{(t-1)}$ is

median value of the texture feature. The number and the type of features can change according to the application at hand.

3.3.2.1 Object Descriptors

Higher level knowledge of the scene can be derived by analyzing the low-level descriptors. The knowledge is derived by clustering region descriptors, thus leading to *semantic descriptors*. In our work, semantic descriptors are *object descriptors*. Each object is processed separately and is decomposed into a set of nonoverlapping regions.

Moving objects are composed of connecting moving regions. Each region descriptor is associated to the corresponding object $O_i^{(t)}$. After this association, the region descriptor is denoted with $\boldsymbol{\varphi}_{i,c}^{t}$. This operation can be expressed as

$$\forall O_i^{(t)},\ i = N_F^{(t)} \qquad \exists\, \boldsymbol{\varphi}_{i,c}^{t} \quad c = 1,\ldots,N_c^{(t)} \tag{3.29}$$

where $N_F^{(t)}$ is number of video objects in frame *t*, and $N_c^{(t)}$ is number of regions for object $O_i^{(t)}$. An example is depicted in Figure 3.7. In this example, there are three objects. Object descriptor is $O_1 = \{\boldsymbol{\varphi}_{1,1}\ \boldsymbol{\varphi}_{1,2}\ \boldsymbol{\varphi}_{1,3}\ \boldsymbol{\varphi}_{1,4}\ \boldsymbol{\varphi}_{1,5}\}$.

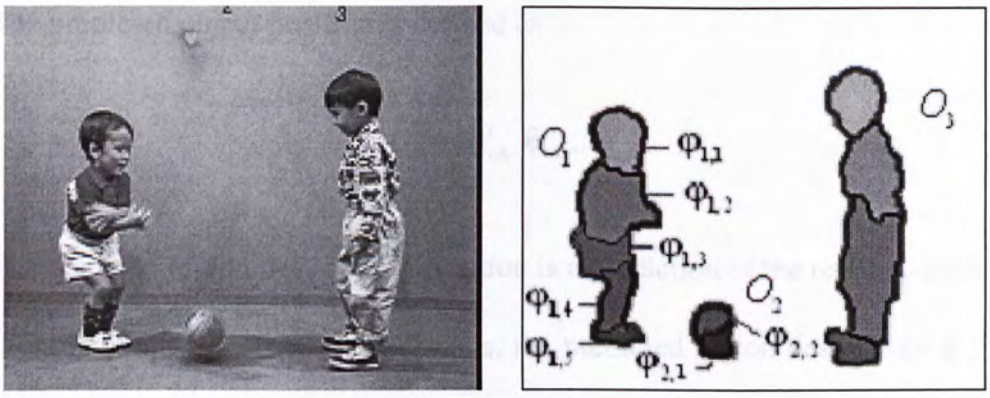


Figure 3.7 An example of object descriptor

3.3.2.1 Region-Object Tracking

The correspondence of video objects in successive frames is achieved through the correspondence of objects' regions. The region-object tracking mechanism is to define a correspondence between the object partition in the previous frame and the object partition in current frame.

The moving region's position predicted through motion compensation is given by

$$\begin{cases} \hat{\varphi}_{r,1}^{(t)} = \varphi_{r,1}^{(t-1)} + \varphi_{r,3}^{(t-1)} \\ \hat{\varphi}_{r,2}^{(t)} = \varphi_{r,2}^{(t-1)} + \varphi_{r,4}^{(t-1)} \end{cases} \tag{3.30}$$

The predicted region descriptor, $\hat{\boldsymbol{\varphi}}_r^{(t)}$, retains the value of the other features unchanged from frame to frame , so that

$$\hat{\boldsymbol{\varphi}}_{i,r}^{(t)} = (\hat{\varphi}_{r,1}^{(t)}, \hat{\varphi}_{r,2}^{(t)}, \varphi_{r,3}^{(t-1)}, \varphi_{r,4}^{(t-1)}, \ldots, \varphi_{r,l}^{(t-1)})^T \tag{3.31}$$

The predicted object position is defined as

$$\hat{O}_i^t = \{\hat{\varphi}_{i,1}^t, \hat{\varphi}_{i,2}^t, \ldots \hat{\varphi}_{i,N_c^{(t)}}^t\} \tag{3.32}$$

The result of region descriptor projection is a prediction of the region partition in the next frame. In order to track regions, the predicted region descriptors $\hat{\varphi}_{i,r}{}^{(t)}$ of the regions belonging to moving objects $O_i^{(t-1)}$ in frame t-1 are compared with the region descriptors φ_c^t of regions in frame t. The region, which has the minimum distance with the region descriptor $\hat{\varphi}_{i,r}{}^{(t)}$, should be marked in the Object $O_i^{(t)}$.

$$\min_C d(\hat{\varphi}_{i,r}{}^{(t)}, \varphi_c^{(t)}) = \min_C \sum_{l=1}^{f} w_l \left| u_{r,l} - u_{c,l} \right| \tag{3.33}$$

where $u_{r,l}$ (respectively, $u_{c,l}$)is the median of the feature u_l in region $R_r^{(t-1)}$ (respectively, R_c^t) and w_l is the weight assigned to the corresponding feature. As shown in Fig 3.8, the region descriptor $\hat{\varphi}_2{}^{(2)}$ will find the corresponding region R_2 for the minimum distance between $\hat{\varphi}_2{}^{(2)}$ and $\varphi_2{}^{(2)}$.

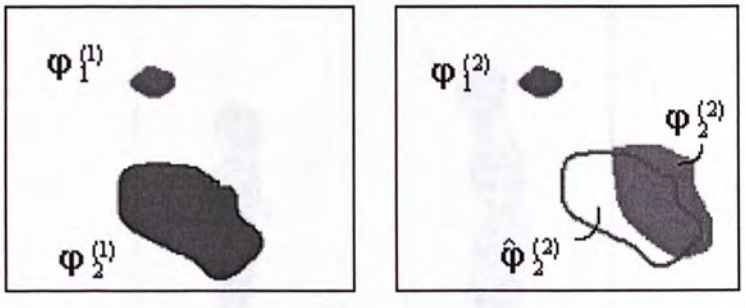


Figure 3.8 An example of region descriptor tracking

A *new object* is detected when a connected set of pixels $S(t)$ belonging to Change Detection Mask but does not get any region descriptor from the projection mechanism. The detection of a new object triggers a track initiation (3.29).

An *occlusion* takes place when two or more object interact, either by getting close one to each other, or by passing one in front of the other. An occlusion is detected when a connected set of pixels $S(t)$ receives projected region descriptors from several objects. The object partition validation step separates the objects, that is, provides separate contours for each different object. This refinement is made possible by using the knowledge of the track at the region level.

A *splitting* corresponds to the separation of a connected set of pixels in the object partition into two or more subsets. This event is detected when two different disconnected sets of pixels $S_1(t)$ and $S_2(t)$ get region descriptors projected from the same video object. At this time, a new object is detected and will be tracked in the following frames.

Figure 3.9 Segmentation result of *Children* sequence.

3.4 Results and Discussions

Several experiments were carried out on different standard video sequences in quarter-common intermediate format (QCIF) to test the performance of this VOP segmentation algorithm. Both the objective and subjective quality evaluations are applied to our algorithm.

3.4.1 Objective Evaluation

The error rate of the object mask is adopted to demonstrate the accuracy of the proposed algorithm. The error rate is defined in [SY02] as follows:

$$\text{Error Rate} = \frac{\text{Error Pixel Count}}{\text{Frame Size}} = \frac{S^t_{VOP} \oplus S^t_{Mask}}{N} \tag{3.34}$$

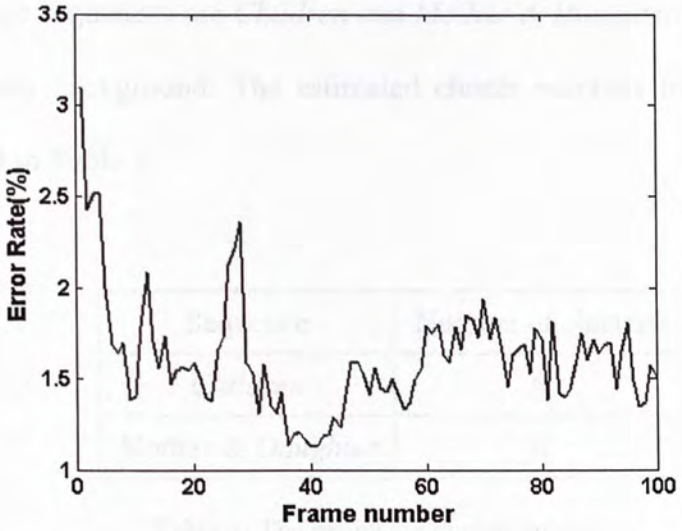


Figure 3.10 Error rate in each frame of the *Children* sequence (QCIF)

where the error pixel count is the number of pixels at which the obtained object mask S^t_{VOP} is different from the ideal alpha plane S^t_{Mask}, that comes with the test

sequence. The alpha plane contains only foreground/ background information and $\oplus$ is logic XOR operation.

Figure 3.10 shows the error rate curve for the *Children* sequence. The x-axes shows the frame number and the y-axes the corresponding error rate of each frame. From the above figure, the error rate is lower than 2% most of time, except at the beginning several frames and some frames near frame 28. The reason is that at the beginning, some parts of objects have no motion, several frames are needed to complete the extraction of foreground object. The sudden rise of error rate about frame 28 corresponds to a large motion of the objects.

3.4.2 Subjective Evaluation

Figures 3.11 and 3.12 show the segmentation results for several benchmark sequences. These sequences are *Children* and *Mother & Daughter*. Both sequences have a stationary background. The estimated cluster numbers for the sequences tested are listed in Table 1.

Sequence	Number of clusters
Children	5
Mother & Daughter	8

Table 1: The estimated cluster number

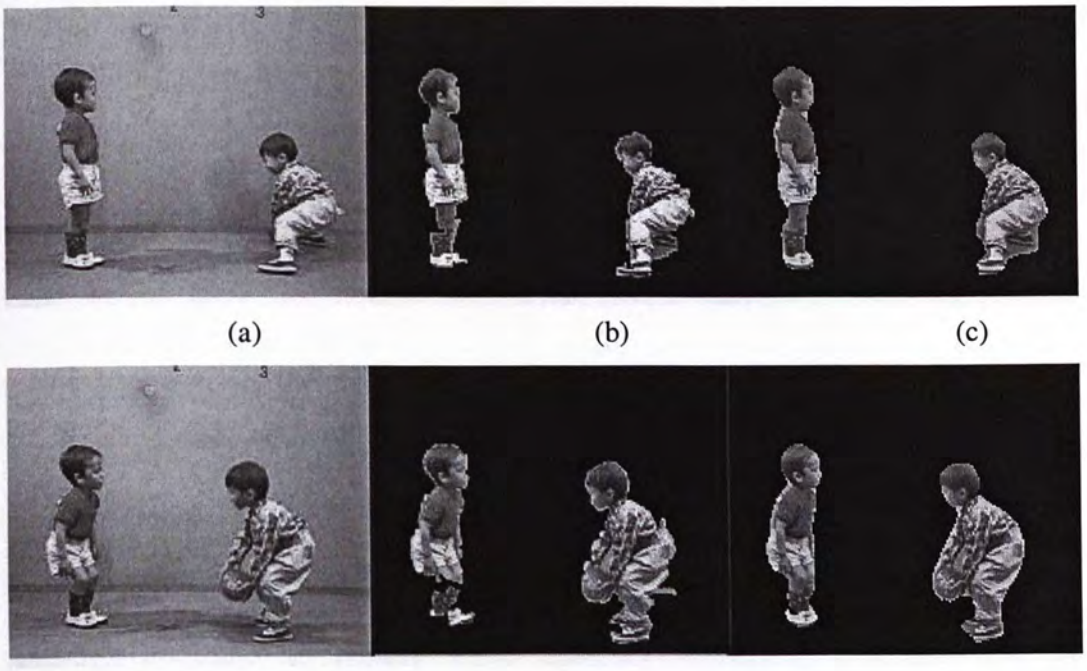
For the proposes of illustration and comparison, we compare the segmentation results of the *Children* and *Mother and Daughter* sequence using the Chien's

algorithm [SY03] and our proposed algorithm. The *Children* sequence is more difficult sequence to be segmented because of the multiple objects and the fast moving shadows on the background. Figure 3.11 (b), Figure 3.11(e) and Figure 3.11 (h) are results of Chien's algorithm, Figure 3.11 (c), Figure 3.11 (f) and Figure 3.11 (i) are results of our proposed algorithm, respectively. As seen in the comparison, our algorithm provides more precise segmentation results, the shapes of children are more complete and the ball is also extracted, which is one of the moving objects and should not be lost.

In the *Mother & Daughter* sequence, the head of the mother has a relatively large motion, while her body exhibits little motion. The motion of the daughter is less throughout the sequence. Our algorithm is still capable of determining the locations of the moving objects reasonably well, as demonstrated in Figures 3.12 (c), Figure 3.12 (f) and Figure 3.12 (i). Comparing to Chien's algorithm, the boundaries of objects are also smoother. The difference in accuracy can be easily observed around the face of the daughter.

Since different foreground objects are to be detected and tracked in the sequence, several frames may be needed to extract all the foreground objects. The foreground objects are fully extracted in frame 3 of the *Children* sequence and frame 12 of the Mother & Daughter sequence.

We also show some results of other sequences. *Claire* is a simple sequence with uncluttered, stationary background. As seen from Figure 3.13, the VOPs contour produced by our segmentation algorithm is reasonable. Figure 3.14 depicts the extracted VOP for *Silent* sequence. The background in this sequence is cluttered in contrast to *Claire*. The boundary location for this more complicated sequence is not as accurate as for the previous sequence *Claire*. Since the background is cluttered, the model identified small parts of the background as foreground. The sequence *Grandmother*'s results are shown in Figure 3.15. In this sequence, because the moving part is only the head of the woman and we only extract moving object, only head is segmented. Figure 3.16 is the results of *Table tennis* sequence, which proves that our algorithm can successfully extract small object even like ping-pong ball.



(a) (b) (c)

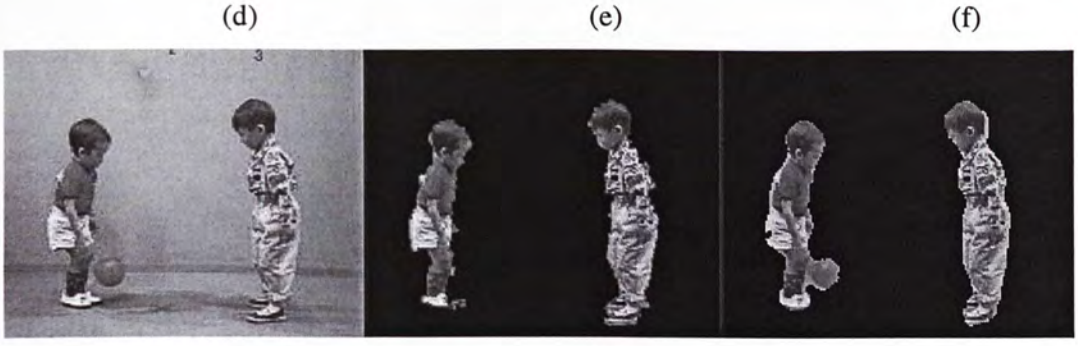
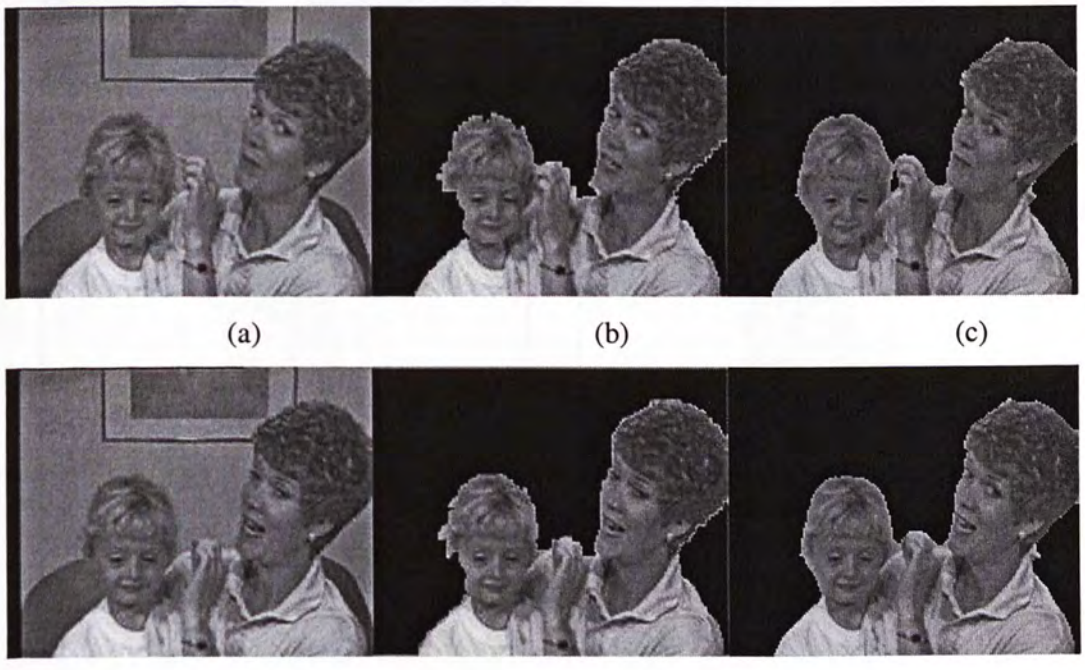


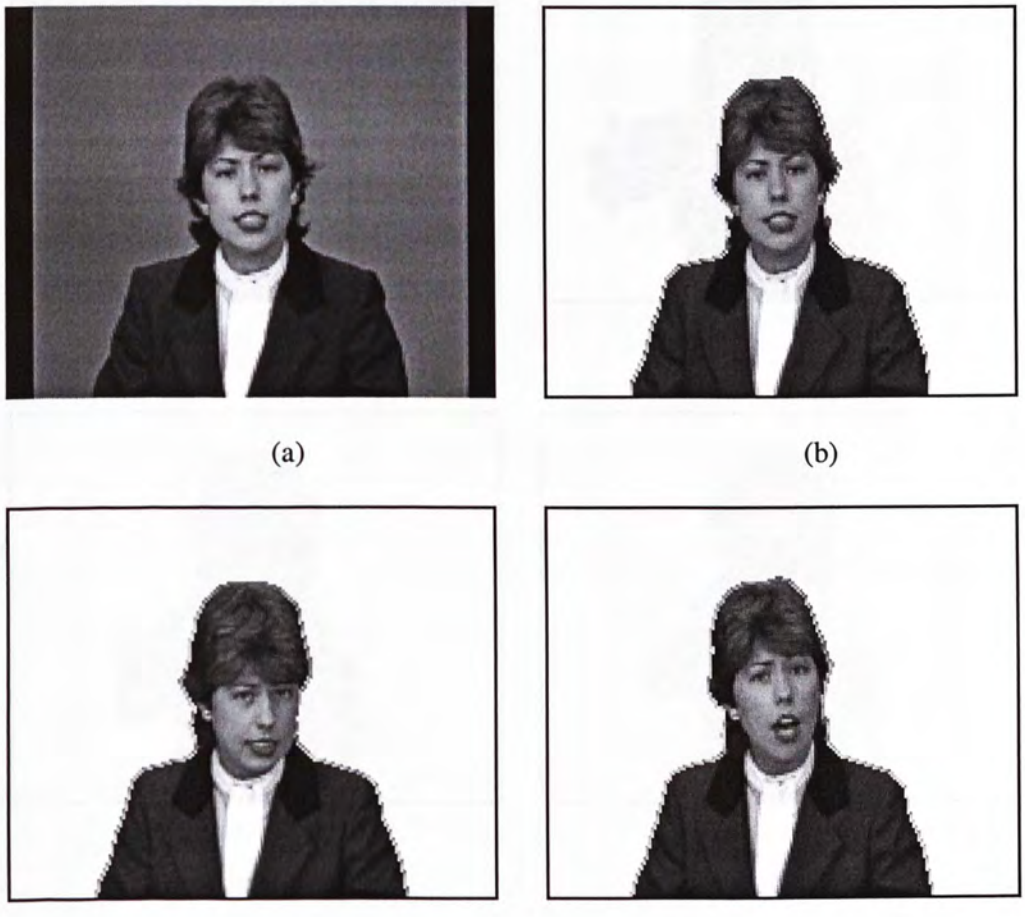
Figure 3.11 The *Children* sequence: (a), (d) and (g) the original frame; (b), (e) and (f) the segmentation results of the Chien's algorithm; (c), (f) and (i) the segmentation results of the proposed algorithm using region descriptor tracking.



(d) (e) (f)

(g) (h) (i)

Figure 3.12 The *Mother & Daughter* sequence: (a), (d) and (g) the original frame; (b), (e) and (f) the segmentation results of the Chien's algorithm; (c), (f) and (i) the segmentation results of the proposed algorithm using binary model tracking.



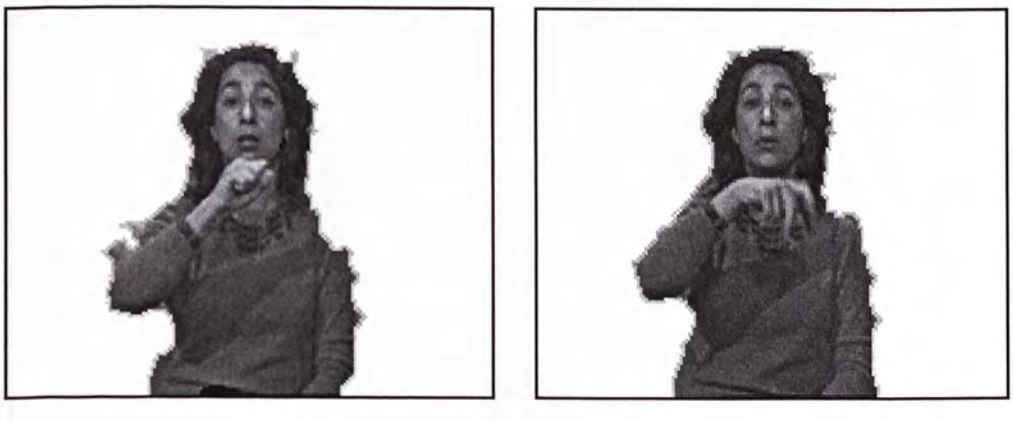
(a) (b)

(c) (d)

(e) (f)

Figure 3.13 Segmentation results of the *Claire* sequence: (a) original frame, (b) ~ (f) segmentation results using binary model tracking.

(a) (b)



(c) (d)

(e)

(f)

Figure 3.14 Segmentation results of the *Silent* sequence: (a) original frame, (b) ~ (f) segmentation results using binary model tracking.

(a)

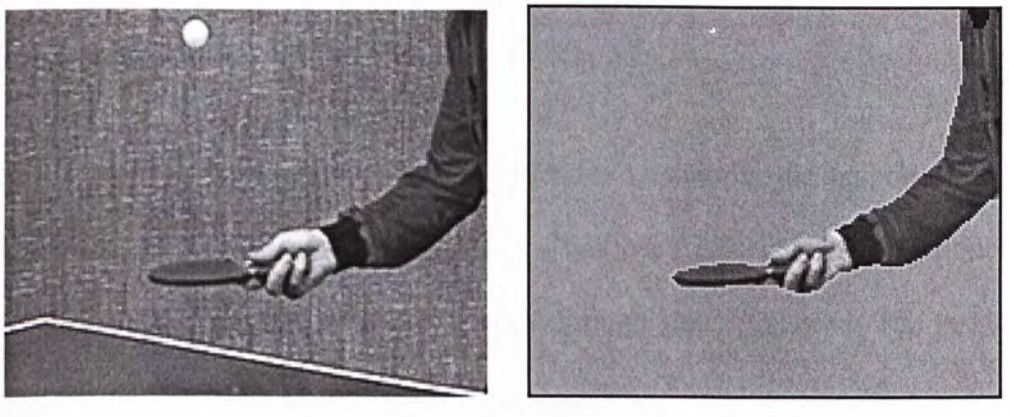
(b)

(c)

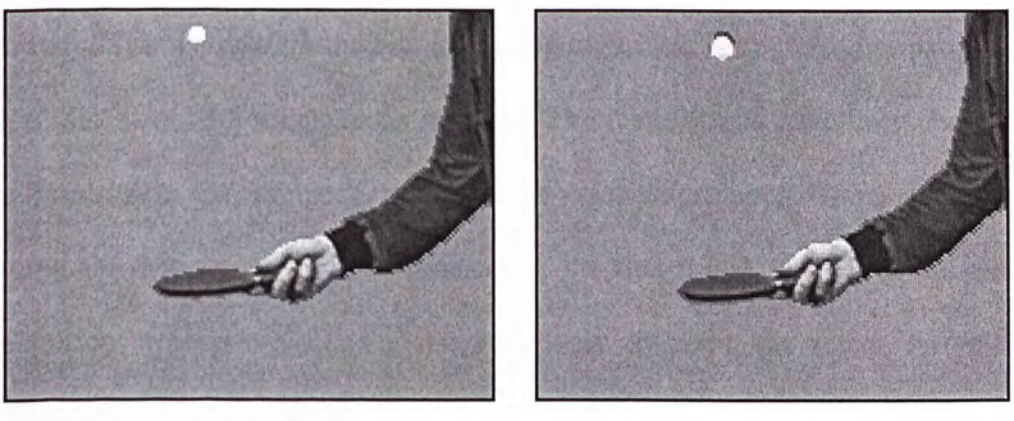
(d)

(e) (f)

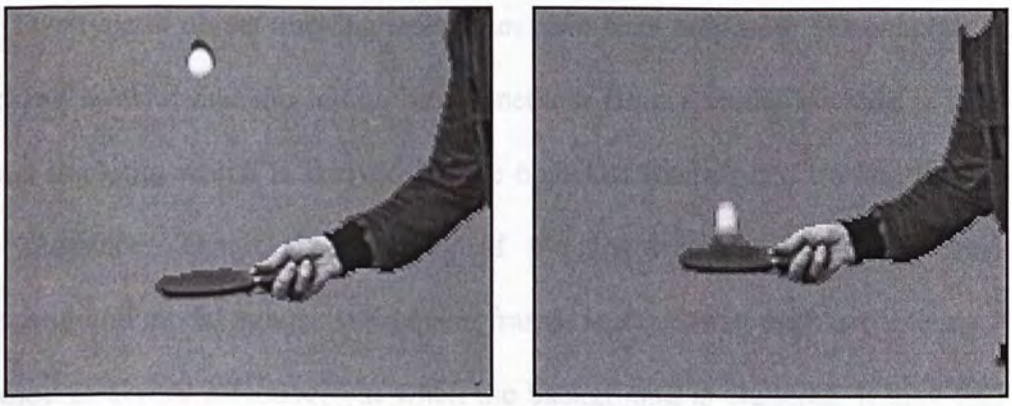
Figure 3.15 Segmentation results of the *Grandmother* sequence: (a) original frame, (b) ~ (f) segmentation results using binary model tracking.



(a) (b)



(c) (d)



(e) (f)

Figure 3.16 Segmentation results of *Table-Tennis* sequence: (a) original frame, (b) ~ (f) segmentation results using region descriptor tracking.

3.5 Conclusion

In this chapter, an automatic VOPs generation method for the support of object-based coding in the framework of MPEG-4 has been presented that continuously separates moving objects in image frames through time evolution. The proposed method utilizes temporal information for localizing moving objects, and spatial information for the acquisition of precise object boundaries and semantic region partitions. The weighted *k*-median clustering algorithm is employed to partition an image into a set of homogeneous regions. The features that have been considered were luminance, chrominance, texture and motion information. The weights of the features are determined by the variances in the first frame, except for the motion information, whose weight is given adaptively by using the region information obtained in the previous frame

Two type of object tracking techniques have been proposed: The template based tracking method and the region based method. Binary model tracking is template based tracking, which is derived for the object of interest and tracked throughout the sequence. Temporal continuity of the segmentation is accomplished by matching the model against subsequent frames and updating them accordingly. This method is a very effective, but when the background is cluttered, it may find the wrong boundaries which are difficult to correct. Region descriptor tracking is a method based on regions. Comparing these two methods, region descriptor tracking can cope with deformation and complex motion, but its computation is much higher than the binary model tracking. Since in every frame, region partition is needed, that is a time consuming process.

Experimental results demonstrate that the proposed method is able to successfully extract moving objects from the sequence. As part of our work, we intent to separate multiple objects using depth information, which would enable a multi-layered representation of a video frame.

Chapter 4
Disparity Estimation and its Application in Video Object Segmentation

Although most video archives mainly consist of 2-D video sequences, the use of 3-D video, obtained by stereoscopic or multiview camera systems, has recently increased since it provides more efficient visual representation and enhances multimedia communication. 3-D video enables users to handle and manipulate video objects more efficiently by exploiting, for example, depth information provided by stereo-image analysis. Furthermore, the problem of content-based segmentation is addressed more precisely since video objects are usually composed of regions belonging to the same depth plane [LF97]. Various applications, such as video surveillance, image/video indexing and retrieval, or editing of video content, can gain from such 3-D representation. For this reason, 3-D data acquisition and display systems have attracted great interests recently and consequently archives of 3-D video information are expected to rapidly increase in the forthcoming years.

Depth information would enable a multi-layered representation of a video frame. Each layer would contain image regions that are at a specific distance from the video camera. The depth information of a scene is contained in a so-called depth map. An example of a depth map is shown in Fig. 4.1. Fig. 4.1(a) shows a group of people sitting at different distances from the camera, and the corresponding depth map is shown in Fig. 4.1(b). Each person is indicated by a different grey-level in the depth map. Unfortunately,

the depth map cannot be employed to extract each person from the image on its own, because other objects (e.g., the glasses and table) that are in the same depth layer would also be extracted. Although depth information would provide a powerful cue in video segmentation, there are certain limitations that must be overcome. We will investigate how depth information can be incorporated into existing video segmentation techniques. This would pave the way to developing novel techniques that will provide a significant improvement over existing methods.

(a) (b)

Figure 4.1 An Example of a disparity map. (a) Original image, (b) depth map

Disparity is defined as the relative displacement between the left and the right image points belonging to the same object point, as defined in Fig. 4.2. It depicts the relative depth information, but compared with depth estimation, it is more easily obtained. Without first needing camera calibration, it will find more applications in generic video communications.

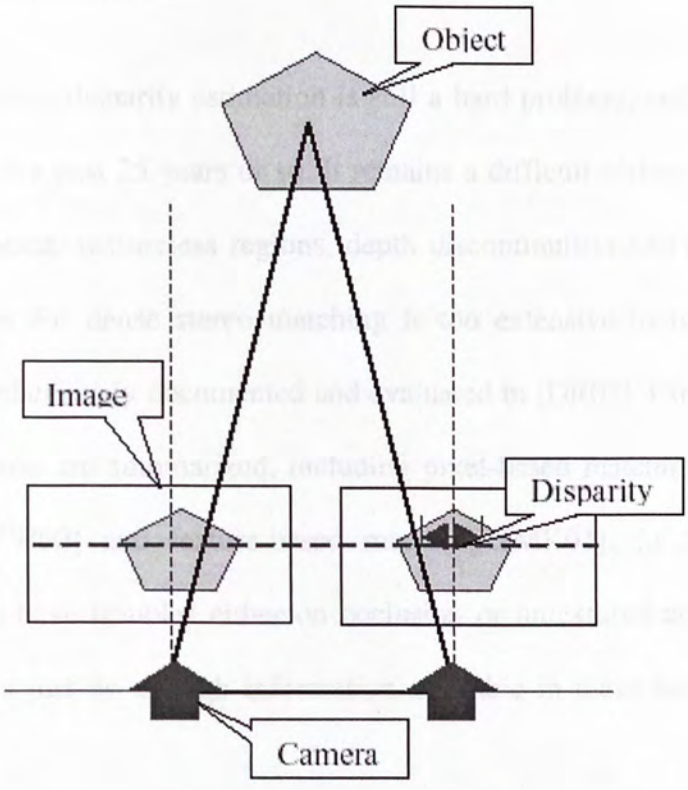


Figure 4.2 Definition of Disparity

In this chapter, we present a disparity estimation algorithm using edge matching and spatial interpolation algorithm on color information and its application in video object segmentation. The remainder of this chapter is organized as follows. Section 4.1 is concerned with disparity estimation using the edge matching and spatial interpolation. Disparity applications in video object segmentation are discussed in Section 4.2, while the Section 4.3 gives a summary of the work.

4.1 Disparity Estimation

It is well known that disparity estimation is still a hard problem, and much progress has been made over the past 25 years or so. It remains a difficult vision problem for the following reasons: noise, textureless regions, depth discontinuities and occlusions. The variety of algorithms for dense stereo matching is too extensive to list here but has recently been comprehensively documented and evaluated in [DR02]. Current techniques for disparity estimation are summarized, including pixel-based matching [IJ96], block-based matching [WW00] and feature-based matching [MK01]. In fact, almost all matching algorithms have troubles either on occlusion or untextured areas. This is not surprising as there is just no enough information available in these areas which allow decision making.

The objective of the proposed disparity estimation algorithm is to provide multi-layer representation of the current frame for object segmentation. Each layer should have smooth boundary in order to obtain a natural appearance of the object. In contrasts to our proposed method, current disparity estimation algorithms usually do not produce a smooth boundary. The basic idea of our proposed edge match propagation algorithm is the same as that in [ML02]. In [ML02], Lhuillier and Quan proposed an efficient match propagation algorithm which produces a quasi-dense pixel matching between two images using region growing principle. The proposed algorithm further develops the idea by starting from a set of sparse matches as seed points, and then propagating the results to edge points to obtain the edge disparity map. After that, an image interpolation method is proposed to obtain the disparity map based on spatial correlation. There are two steps in

this algorithm. The first step is to find a set of corresponding points and propagate them to edge pixels. The second step is to interpolate the rest of the pixels with disparity values.

4.1.1. Seed Selection

In the proposed algorithm, a pair of images can be captured by a hand-held camera or a stereo camera. The first image is denoted by I_1 and the second is I_2. In our implementation, all the corner points obtained by Susan corner detector [SM97] are selected as feature points, and here the edge maps are obtained using Canny operator [JC86]. For feature point p_1 in I_1, we use a correlation window of size $(2n+1)\times(2m+1)$ centered at this point, and then select a rectangular search area of size $(2d_x+1)\times(2d_y+1)$ around this point in I_2. A correlation operation is performed on a given window between p_1 and all feature points p_2 lying within the search area in I_2. The correlation score is defined in [ZZ95] as

$$Score(p_1,p_2)=\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}[I_1(x_1+i,y_1+j)-\overline{I_1(x_1,y_1)}]\times[I_2(x_2+i,y_2+j)-\overline{I_2(x_2,y_2)}]}{(2n+1)(2m+1)\sqrt{\sigma^2(I_1)\times\sigma^2(I_2)}} \tag{4.1}$$

where $\overline{I_k(x,y)}=\sum_{i=-n}^{n}\sum_{j=-m}^{m}I_k(x+i,y+j)/[(2n+1)(2m+1)]$ is the intensity average at point (x, y) of I_k (k=1, 2), and $\sigma(I_k)$ is the standard deviation of image I_k in the neighborhood $(2n+1)\times(2m+1)$ of (x, y), which is given by

$$\sigma(I_k) = \sqrt{\frac{\sum_{i=-n}^{n}\sum_{j=-m}^{m}(I_k^2(x,y) - \overline{I_k(x,y)})}{(2n+1)(2m+1)}} \tag{4.2}$$

This correlation measurement does not need to first perform image balancing. Compared with other matching criteria, it is not affected by lighting and camera differences. The score ranges from –1, if the two correlation windows are not similar at all, to 1, for two correlation windows which are identical. For a given couple of points to be considered as candidate match, the correlation score must be higher than a given threshold. In our implementation, the threshold is set at 0.8. For each feature point in the first image, there is a set of candidate matches from the second image; and at the same time, there is also a set of candidate matches from the first image for each feature point in the second image. The method "some-winners-take-all" [ZZ95] is used to update the matching in order to retain a one-to-one matching between two images. This method checks the correlating pixels of the first to the second and inversely by correlating those of the second to the first image, and only the best matches consistent in both directions are retained. Fig.4.3 (c) (d) depicts the one to one matching points in *Flower Garden* stereoscopic image pair.

(a) (b)

(c) (d)

Figure 4.3 *Flower Garden* stereoscopic image pair: (a) original image 1, (b) original image 2, (c) matching points in image 1, and (d) matching points in image 2.

4.1.2. Edge-based Matching by Propagation

All the corresponding points obtained in the previous step are sorted in decreasing correlation score as the seed points for concurrent propagations. At each step, the match (p_1, p_2) composed of two corresponding points p_1 and p_2 with the best correlation score is removed from the current set of seed matches. Then we search for possible new matches in the immediate spatial neighborhood $N(p_1, p_2)$ as defined below.

Let

$$N(p_1) = \{q_1,\ \ q_1 - p_1 \in [-r, r]^2,\ \ q_1 \in E_1\} \tag{4.3}$$

$$N(p_2)=\{q_2,\ q_2-p_2\in[-r,r]^2,\ q_2\in E_2\} \tag{4.4}$$

denote all $(2r+1)\times(2r+1)$ neighboring edge pixels of pixels p_1 and p_2, where E_1 and E_2 are denoted as the sets of all edge pixels detected by the Canny operator in image I_1 and image I_2, respectively. The possible matches limited by the discrete 2D disparity gradient [ML02] are given as

$$N(q_1,q_2)=\{(q_1,q_2),q_1\in N(p_1),q_2\in N(p_2),\|(p_1-p_2)-(q_1-q_2)\|\subseteq\varepsilon\} \tag{4.6}$$

The edge-based propagation algorithm can be described as follows: The input of the algorithm is the set *Seed* of the current seed matches. The set is implemented with heap data structure for both the fast selection of the best match and the incremental addition of seeds. The output is an injective edge disparity *Map*.

Input: *Seed*

Output: *Map*

Map←∅

Initialize the threshold $z = 0.8$

While Seed ≠∅

pull the correlation score best match (p_1,p_2) from *Seed*

Local←∅

for each (q_1,q_2) in $N(p_1,p_2)$ do

if *Score* $(q_1,q_2) > z$

then store (q_1,q_2) in *Local*

end-if

end-for

while *Local* $\neq\varnothing$

pull the correlation score best match (q_1,q_2) from *Local*

if $(q_1,*)$ and $(*,q_2)$ are not in *Map*

then store (q_1,q_2) in *Map* and *Seed*

end-if

end-while

end-while

Figure 4.4 Edge disparity map of *Flower Garden* image pair

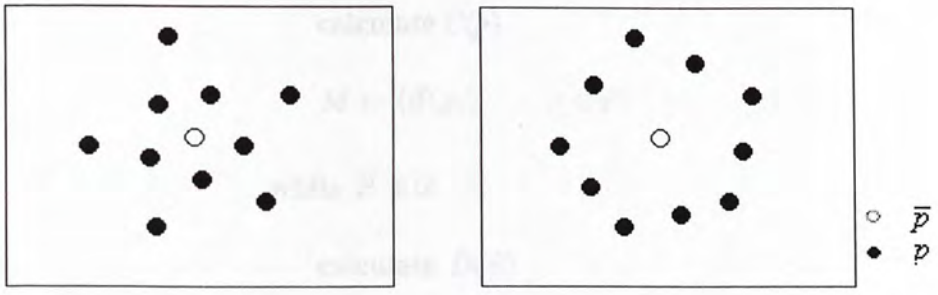
4.2 Remedy Matching Sparseness by Interpolation

We use a new image interpolation method based on spatial correlation to remedy matching sparseness. This algorithm can handle untextured regions and occlusion areas, which present a challenge to many existing stereo algorithms. At the same time, it does not produce undesirable artifacts as some traditional interpolation algorithms do. The

points whose disparity values are dubious are called disputed points. Let $\overline{P}$ be the set of points whose disparity cannot be found, $\overline{P} = \{\overline{p}_1, \overline{p}_2, ..., \overline{p}_k\}$, and P be the set of points whose disparity values are already obtained, $P = \{p_1, p_2, ..., p_l\}$. The disparity of pixel p_i is defined as $\delta(p_i)$. This interpolation algorithm starts from points in $\overline{P}$ which have high density and low color changes. The density $D(\overline{p})$ of pixel p is defined as

$$D(\overline{p}) = \sum_{i=1}^{m} f(\overline{p}, p_i) = \sum_{i=1}^{m} \frac{1}{\sqrt{2\pi}} e^{-\frac{(\overline{p}-p_i)^2}{2\sigma^2}} \tag{4.7}$$

The reason of using Gaussian function to calculate density is that to pixel "$\overline{p}$", the more pixels "p" are nearby, the higher $D(\overline{p})$ should be, which is shown in Fig. 4.5. In Fig. 4.5 (a) and (b), the numbers of "p" are the same, but the densities of "$\overline{p}$" are quite different. The color change $C(\overline{p})$ is calculated by color standard deviation over a small window centered at the pixel "$\overline{p}$". For each "$\overline{p}$", the interpolation disparity is more reliable if it is calculated from points "p" of the same region. If the color information of points "p" and "$\overline{p}$" are more similar, it is more perceivable that they are from the same region. Therefore, "$\overline{p}$" with low $C(\overline{p})$ is preferred to be interpolated first. Also, interpolation is more reliable if it is calculated from nearby "p". So "$\overline{p}$" with more nearby "p" should be selected to be interpolated first. Therefore high $D(\overline{p})$ is preferred. We propose to use $D(\overline{p})$ / $C(\overline{p})$ as the selection criterion. The point with the highest $D(\overline{p})/C(\overline{p})$ is selected first for interpolation.



(a) (b)

Figure 4.5 Density map of pixel "$\overline{p}$"

In the interpolation, again we prefer to use points "p" which have similar color with point "$\overline{p}$". Therefore, weights are given to different "p" based on the color similarity to the current "$\overline{p}$". The weight is defined as

$$w(\overline{p}, p) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(I(\overline{p})-I(p))^2}{2\sigma^2}} \tag{4.8}$$

The disparity value $\delta(\overline{p})$ of $\overline{p}$ is obtained by

$$\delta(\overline{p}) = \frac{\sum_{p_j \in N(\overline{p})} w(\overline{p}, p_j)\delta(p_j)}{\sum_{p_j \in N(\overline{p})} w(\overline{p}, p_j)} \tag{4.9}$$

where $N(\overline{p})$ is a small window centered at $\overline{p}$.

The interpolation algorithm can be described as follows. The input of the algorithm is the set of disputed pixels P and the set of undisputed pixels $\overline{P}$. The output is the disparity map M.

Input: P and $\overline{P}$

Output: map M

calculate $C(p)$

$M \leftarrow \{\delta(p_i);\quad p_i \in P\}$

while $\overline{P} \neq \varnothing$

calculate $D(\overline{p})$

pull the pixel $\overline{p}$ with largest $D(\overline{p})/C(\overline{p})$ from $\overline{P}$

interpolate the disparity value $\delta(\overline{p})$ of $\overline{p}$

store $\overline{p}$ in P

store $\delta(\overline{p})$ in M

end while

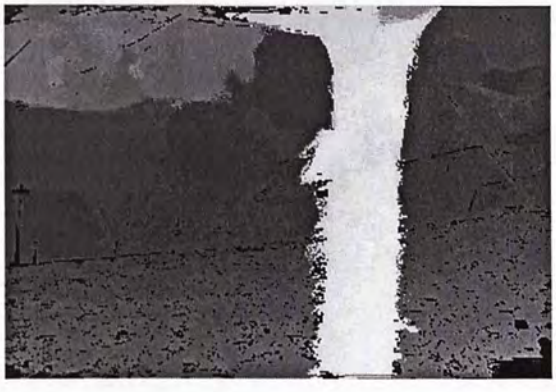
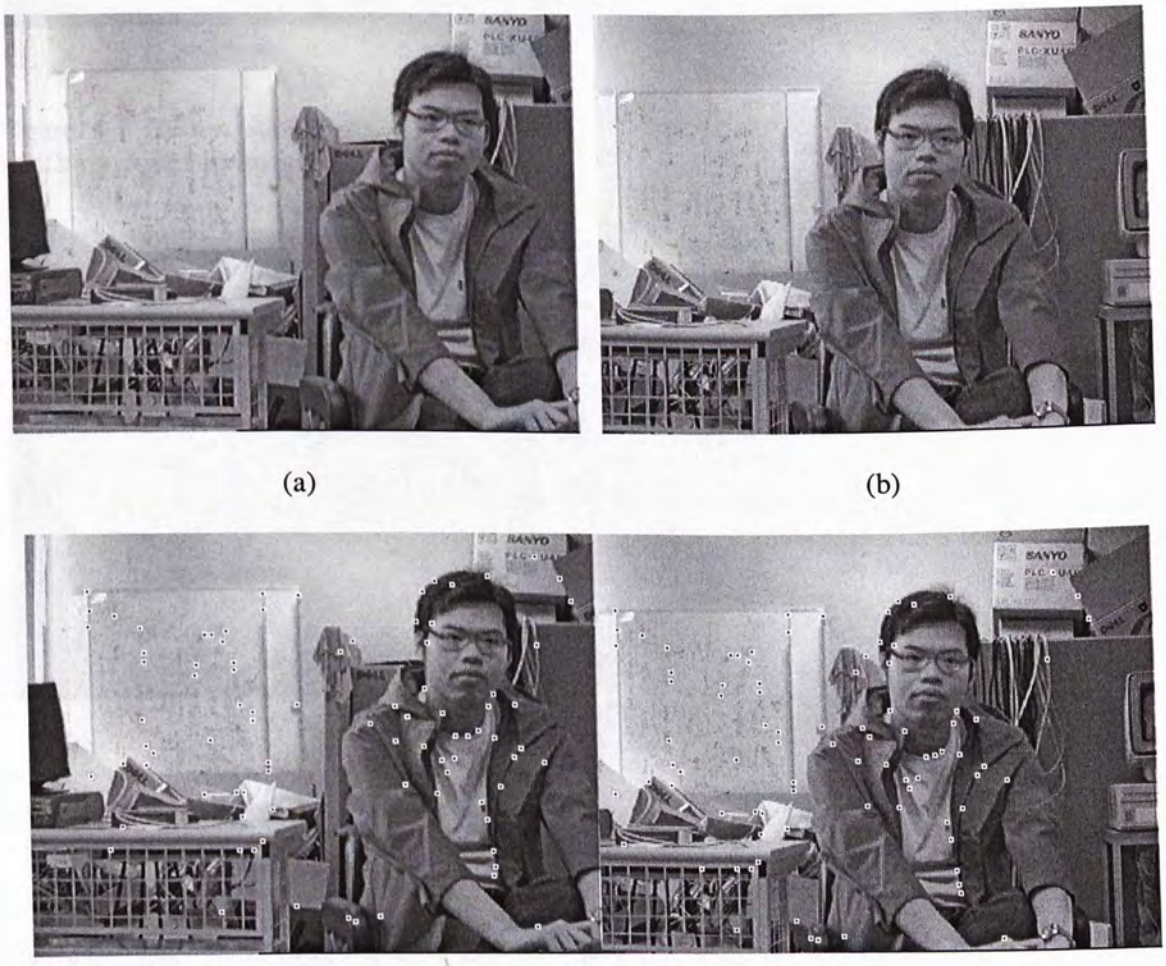


Figure 4.6 Disparity map after interpolation of *Flower Garden* stereoscopic image pair

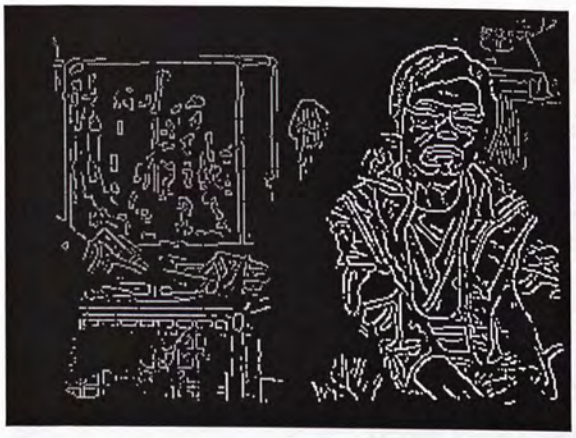
To evaluate the effectiveness of the proposed disparity estimation algorithm, we use different kinds of image pairs as test images. Figure 4.7 (a) (b) is a pair of gray images captured by hand-held camera from two different positions. The PIANO image pairs are typical stereoscopic images, which are shown in Fig. 4.8(a) (b). Image pairs in Figure 4.9 are captured by a stereo camera in our lab and Figure 4.10 (a) (b) is synthesized

stereoscopic image pair. From the disparity maps of all the image pairs, it can be seen that our algorithm is capable of providing an accurate disparity map. All the pixels in the same region will have similar disparities. Compared with other disparity estimation algorithm using block matching, the boundaries of objects have no blocky results.



(a) (b)

(c) (d)



(e)

(f)

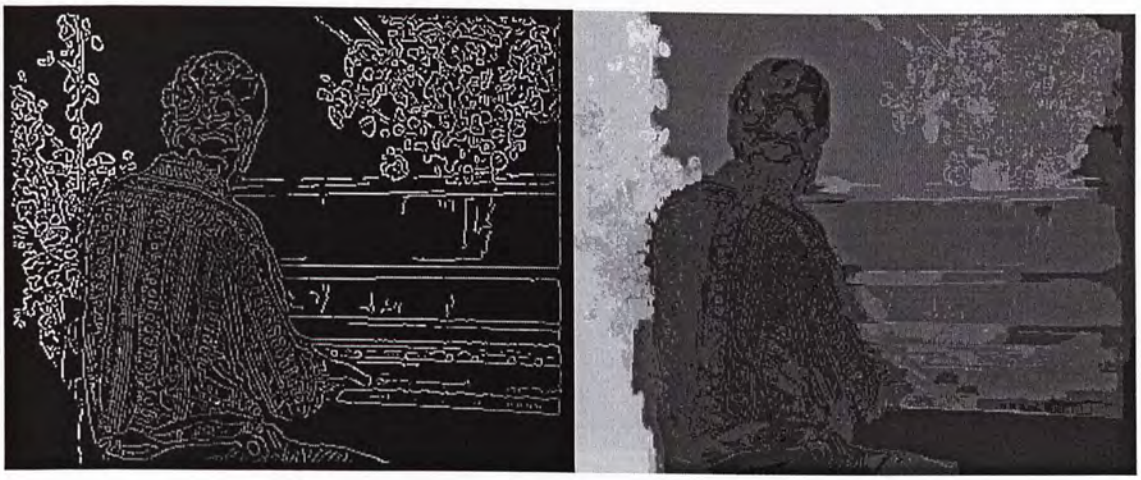
Figure 4.7 *Nelson* stereoscopic image pair: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map.

(a)

(b)

(b)

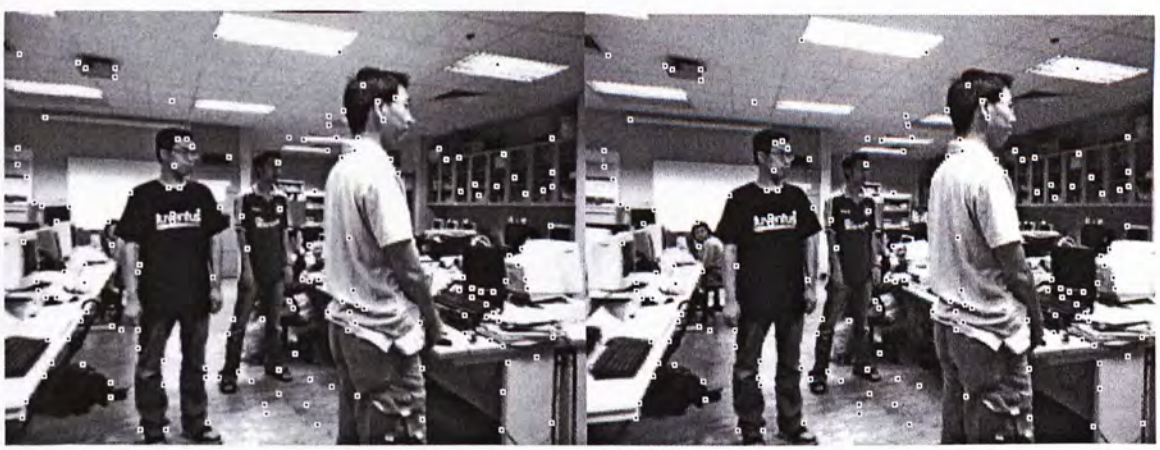
(d)



(e) (f)

Figure 4.8 *Piano* stereoscopic image pair: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map.

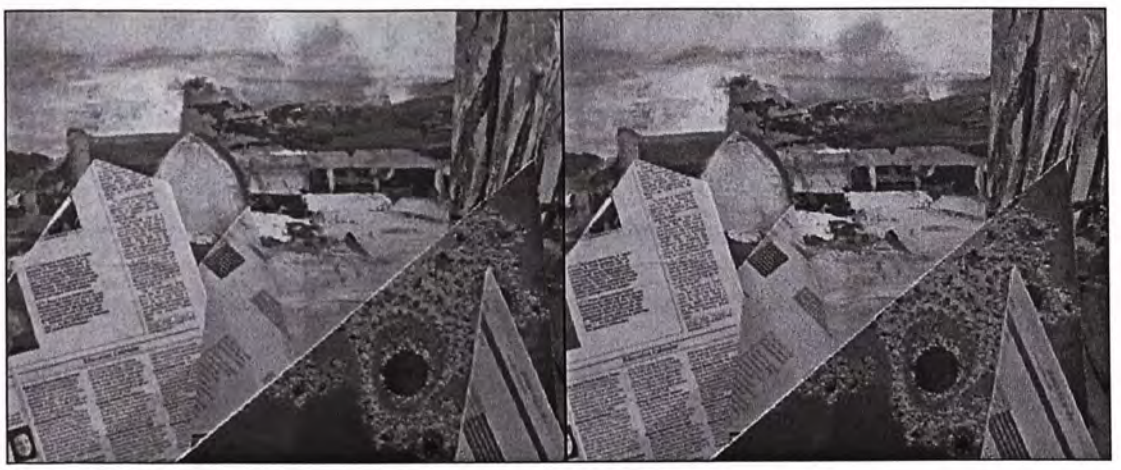
(a) (b)



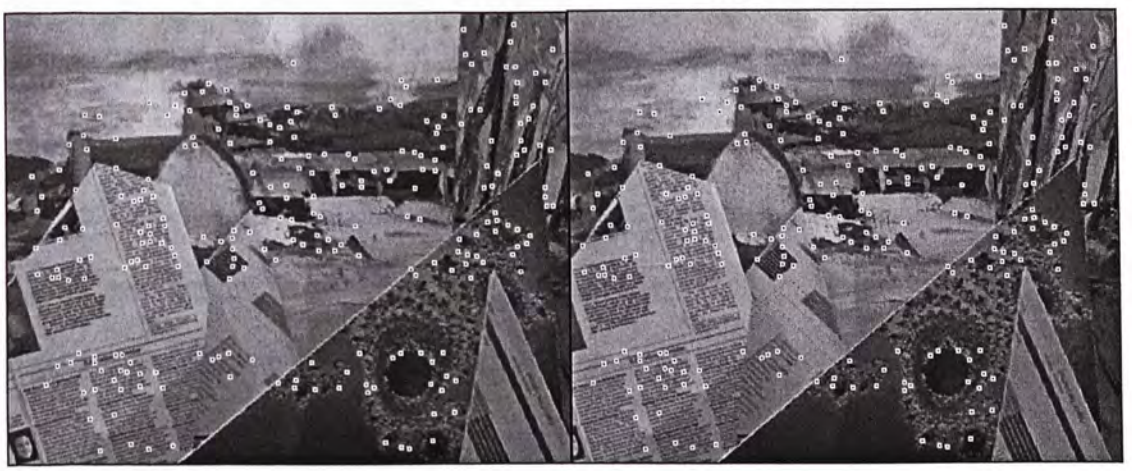
(c) (d)

(e) (f)

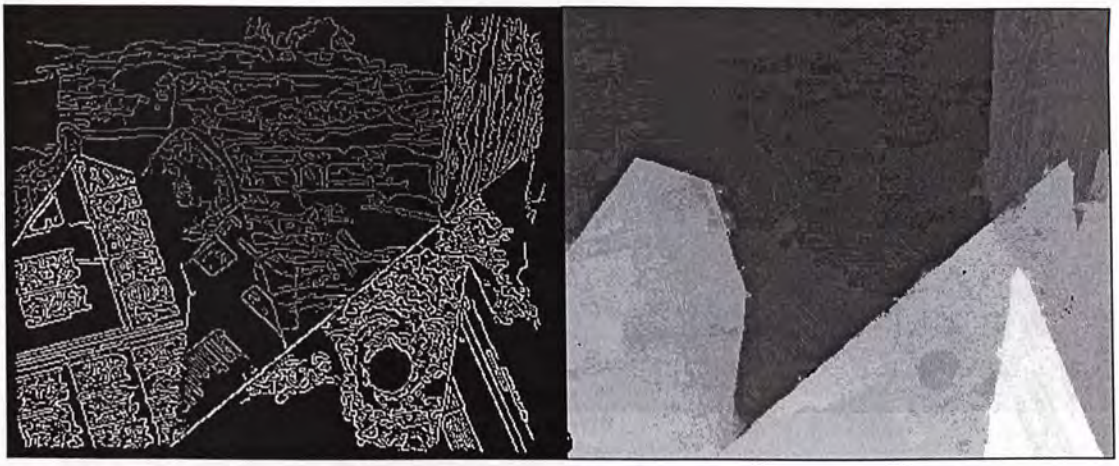
Figure 4.9 *Lab* stereoscopic image pair: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map.



(a) (b)



(c) (d)



(e) (f)

Figure 4.10 Synthesized stereoscopic image pair: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map.

4.2 Disparity Applications in Video Conference Segmentation

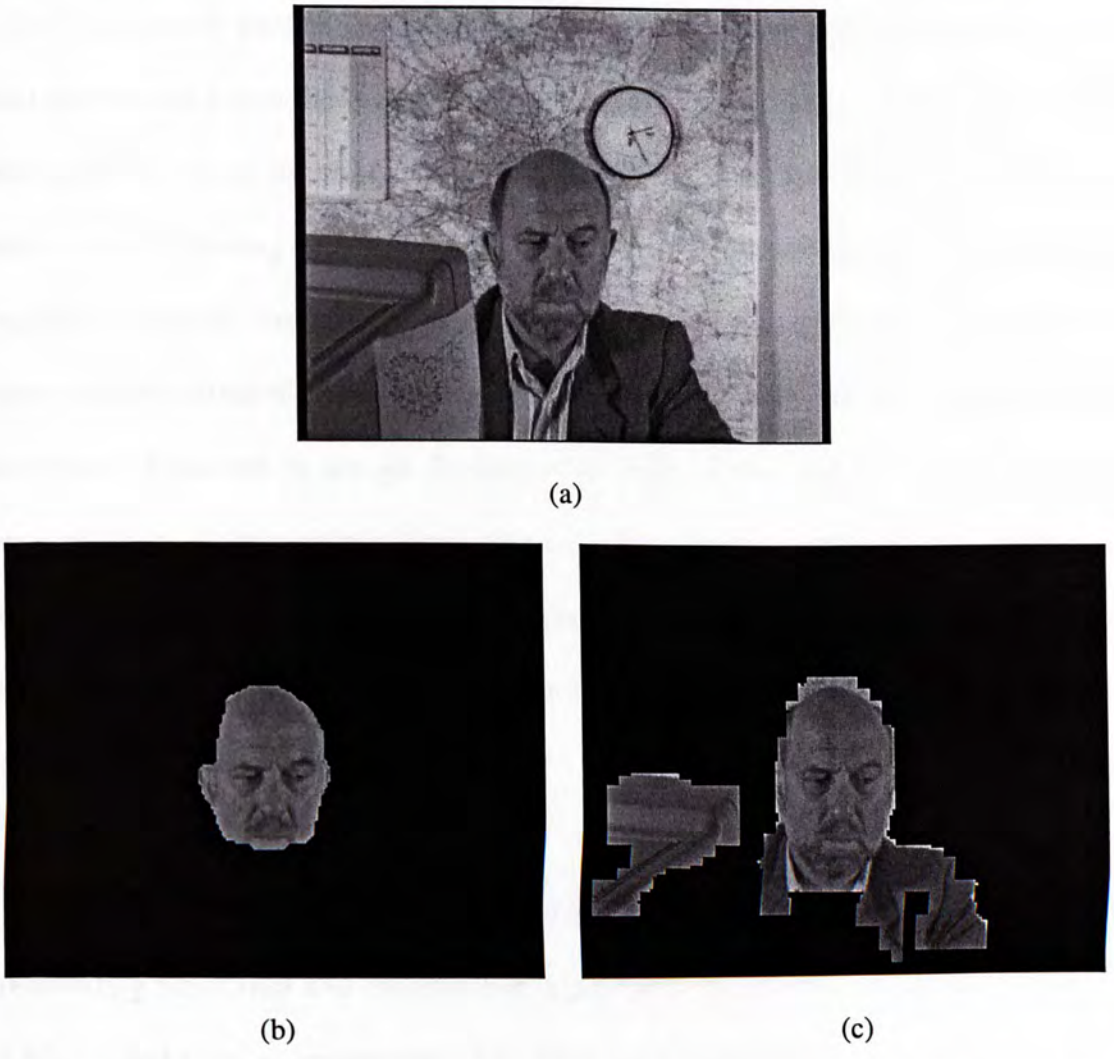
The motivation of segmentation head-and-shoulder type video sequences stems from the popular presence of head-and-shoulder type video signal in real-time services such as videophone and web chatting. In a videoconference sequence, the main object of interest is always the person in front of the camera. Though segmentation of head-and-shoulder sequence is very useful, among the algorithms we reviewed so far, most algorithms are based on face detection and motion information, which are not good enough. Moving object segmentation uses motion information as rule to group regions. The reason is that physical objects are often characterized by a coherent motion, which is different from that of the background. As a result, these kinds of algorithms are restricted to the moving object segmentation, primarily taking advantage of motion information, the weakness of which is not able to handle interesting objects without motion. If the shoulder of human is stationary, it will be lost throughout the sequence. To illustrate this problem, an example

is shown in Fig. 4.11. In *Grandmother* sequence, only the head of the woman has motion. So if use the motion as the cue to extract objects, the body below the head will be lost.

(a) (b)

Figure 4.11 Segmentation results of the *Grandmother* sequence: (a) original frame, (b) segmentation results

Using color information is anther method to detect human. These kinds of algorithms can only detect skin color pixels, like the face and hand parts, which has been discussed in [DC99, NH04, and HL06]. Fig. 4.12 (b) and (c) show the segmentation results of algorithm in [HL06] and [DC99]. But in common sense, human's head and shoulder can not be separated, they should be the parts of one object. So in some applications, like video surveillance, these methods will fail.



(a)

(b) (c)

Figure 4.12 *Claude* stereoscopic image pair: (a) One of the original image pair, (b) segmentation result of [HL06], and (c) segmentation result of [DC99].

Disparity information enables a multi-layered representation of a video frame. Since video objects are usually located on the same depth plane, so depth segmentation provides meaningful frame content representation. The advantage of using disparity is robust to motion fluctuation, even the object stays still for arbitrarily long period of time or when its different parts exhibit different motion characteristics.

The layer which attracts users' interest will be determined using the some criteria. Visual attention is a neurobiological concept having the ability to concentrate the mental power upon an object on close observation. Different applicants may define different attention model. Moving object segmentation approaches assume that objects of interests have distinct motions from background. Photographers always think the most important objects should be located in the center of the image. In a videoconference sequence, the main object of interest is always the person in front of the camera. In this case, the general position of the person and the skin color become very important information in extracting the object (the person's head and shoulder). Face saliency map (FSM) proposed in [HL06], which considered chrominance, luminance and position information, will indicate the human position.

Assume (x, y) represents the spacial position of a pixel in the current image. The corresponding luminance and chrominance components of the pixel are denoted by $Y(x, y)$, $Cb(x, y)$, and $Cr(x, y)$, respectively. The FSM can be defined as

$$FSM(x, y) = P_1(x, y) \cdot P_2(x, y) \cdot P_3(x, y) \tag{4.10}$$

where P_1, P_2, and P_3 denote the "conspicuity maps" corresponding to the chrominance, position, and luminance components, respectively.

Chrominance Conspicuity Map (CCM) P_1: It is known that face region generally exhibits the skin-color feature. Therefore, using the skin-color information, the facial

saliency map can be easily constructed to locate the potential face areas. The P_1 is defined as

$$P_1(x,y) = \exp\{-(\omega_{Cr}(x,y)\frac{Cr'(x,y)^2}{2\Delta_{Cr}^2} + \omega_{Cb}\frac{Cb'(x,y)^2}{2\Delta_{Cb}^2})\} \tag{4.11}$$

$$Cr'(x,y) = (Cr(x,y) - \mu_{Cr})\cos(\theta) + (Cb(x,y) - \mu_{Cb})\sin(\theta) \tag{4.12}$$

$$Cb'(x,y) = (Cb(x,y) - \mu_{Cb})\cos(\theta) + (Cr(x,y) - \mu_{Cr})\sin(\theta) \tag{4.13}$$

where $\mu_{Cr} = 153$, $\mu_{Cb} = 102$, $\Delta_{Cr} = 20$, $\Delta_{Cb} = 25$, $\theta = \frac{\pi}{4}$ and $\omega_v(x,y)$ ($v = Cr$ or Cb) is a weight coefficient.

Position Conspicuity Map (PCM) P_2: In typical head-and-shoulder video sequences, most of the face locations appear at or near the center of the image in order to attract user attention distinctly. Few human faces are captured and placed at the boundary of the image, especially the bottom of the image. Hence, it is reasonable to assume that the probability of the face pixels existing at the center of the image will be larger than other locations. Let H and W denote the height and width of the image, respectively. Based on this characteristic, the Position Conspicuity Map P_2 is defined as

$$P_2(x,y) = \exp\{-\frac{(x - H/2)^2}{0.8\cdot(H/2)^2} - \frac{(y - W/2)^2}{2\cdot(W/3)^2}\} \tag{4.14}$$

Luminance and Structure Conspicuity Map (LSCM) P_3: From the histogram of Y component for the facial test data, the region of [128-50, 128+50] tends to contain most of conspicuity values for the facial skin area. The darker the intensity value of a pixel, the

less possible will be a skin-tone color. Similar result can also be found the very bright pixels. With these observations, LSCM P_3 is defined as

$$P_3(x,y) = s \cdot \exp\{-\frac{(\gamma(x,y) \cdot Y'(x,y) - \mu_L)^2}{2 \cdot \Delta_L^2}\} \tag{4.15}$$

where $\mu_L = 128$, $\Delta_L = 50$, $\gamma(x,y)$ denotes the luminance compensation coefficients. Here s denotes the structural coefficient, which is employed to characterize the luminance variation in face region.

According to (4.10), the final FSM can be easily obtained by employing the three conspicuity maps (4.11), (4.14), and (4.15). For example, the FSM of *Claude* stereoscopic image is computed, and depicted in Figure 4.13(b).

(a)

(b) (c)

Figure 4.13 *Claude* stereoscopic image pair: (a) Original image, (b) FSM, and (c) disparity map

With the observation of above FSM map, it can be found that in the face region, the pixel's FSM value is larger and pixels with large value are more centralized. Combing with the disparity estimation results, the layer representing human tends to contain most of conspicuity values for the facial skin area. Layer L_i represents layer i, when L_i contains the largest conspicuity value S_{Li}, L_i will be 1 and extracted.

$$L_i = \begin{cases} 1, & \text{if } S_{Li} = S_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{4.16}$$

Here S_{Li} denotes the conspicuity value of every layer L_i and defined by the following formula.

$$S_{Li} = \sum FSM(x, y) \times w_f(x, y), \quad (x, y) \in L_i \tag{4.17}$$

$$S_{\max} = \max(S_{Li}) \tag{4.18}$$

where FSM(x, y) is the face saliency value of pixel (x, y), w_f is the a weight coefficient, which is employed to characterize the density of bright pixels in neighboring region of

pixel (x, y). The larger density value and the larger FSM value of pixels, the larger conspicuity value S_{Li} will be. In order to avoid the influence of discrete bright pixel, the weight coefficient w_f is defined as

$$w_f(x,y) = \frac{\sum_{k,l=-r}^{r} B(FSM(x-k, y-l), T)}{r^2} \tag{4.19}$$

$$B(FSM, T) = \begin{cases} 1, & FSM \geq T \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

$$T = \tau \cdot F_{\max} \tag{4.21}$$

$$F_{\max} = \max(FSM) \tag{4.22}$$

where τ is a threshold. According to the performance of a lot of experiments, the value of 0.7 is recommended, which can provide better constraint result for the candidate face pixels selection. Only the pixels with FSM value larger than the threshold will be considered, the pixels with small FSM will not influence the conspicuity value of every layer. When we calculate w_f of pixel (x, y), only $r \times r$ neighborhood of pixel (x, y) are considered.

From (4.17), we can see that only those pixels with larger FSM values and higher weight coefficient will be taken into account, which means that the influence yielded by the discrete pixels with large FSM values and consecutive pixels with small FSM values will be reduced significantly.

Claude stereoscopic image pair is a typical video conference scene. Figure 4.13 (b) is the FSM of stereoscopic image *Claude*, and the layer representing human is depicted in Fig. 4.14(c). To illustrate the effectiveness of our algorithm, more experimental results in Figure 4.15~ 4.17 were obtained by applying the proposed method different kinds of video conference image pairs. It is shown that our method is capable of segmenting the human quite effectively.

(a) (b)

(c)

Figure 4.14 *Claude* stereoscopic image pair: (a) Original image, (b) Disparity map, (c) FSM, (d) layer representing human

(a)

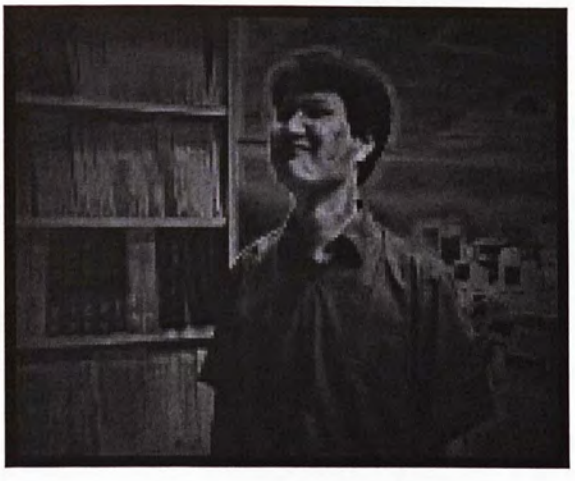
(b)

(c)

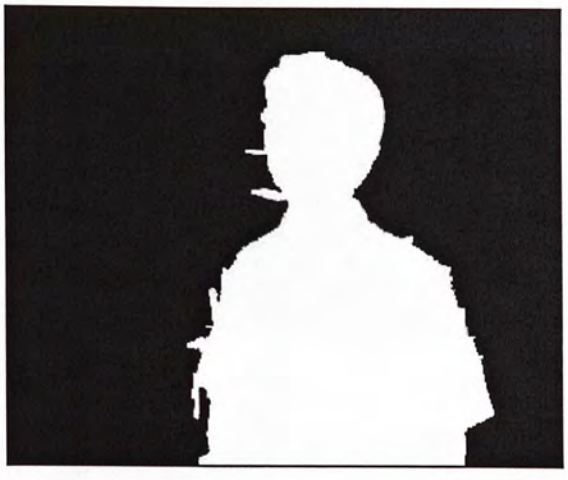
(d)

(e)

(f)



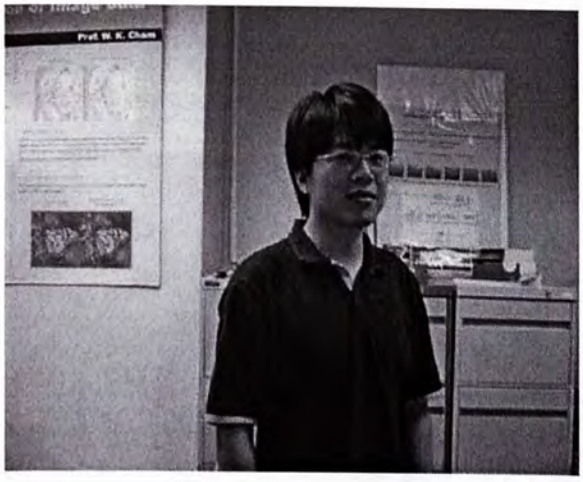
(g)



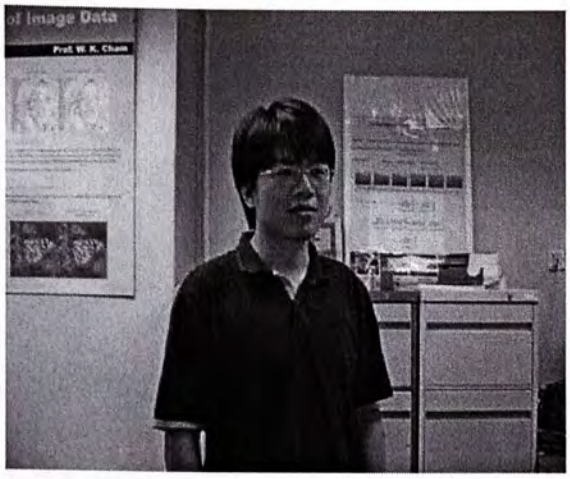
(h)

(i)

Figure 4.15 Image pair captured by hand-held camera: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map, (g) FSM, (h)layer representing human and (i) extracted object.



(a)

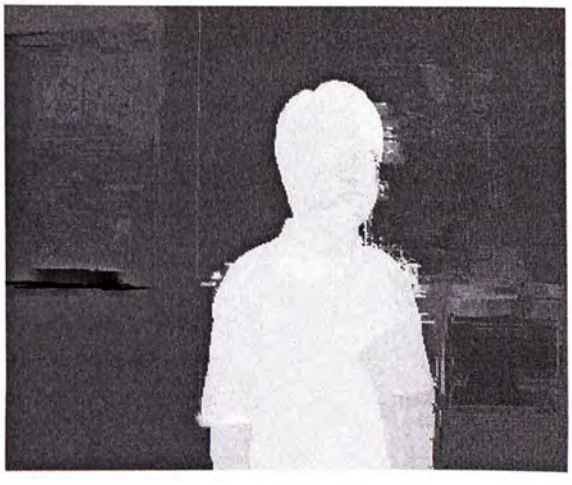


(b)

(c)

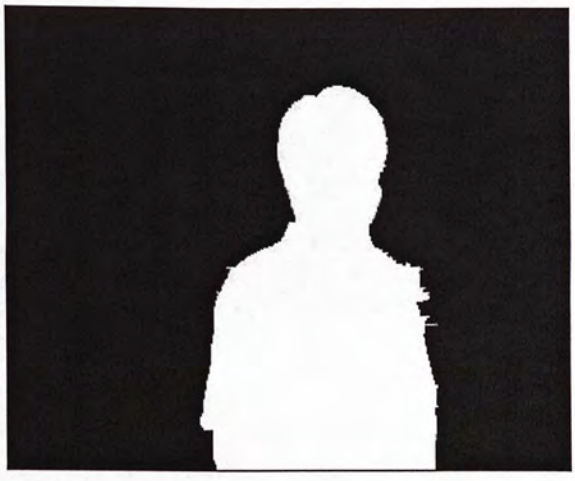
(d)

(e)



(f)

(g)



(h)

(i)

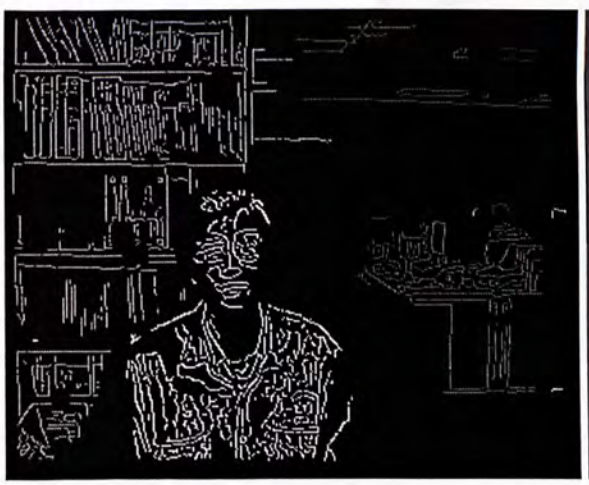
Figure 4.16 Image pair captured by stereo camera: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map, (g) FSM, (h)layer representing human and (i) extracted object.

(a)

(b)

(c)

(d)



(e)

(f)

(g) (h)

(i)

Figure 4.17 Image pair captured by hand-held camera: (a) original image 1, (b) original image 2, (c) matching points in image 1, (d) matching points in image 2, (e) edge disparity map, (f) disparity map, (g) FSM, (h)layer representing human and (i) extracted object.

4.3 Conclusion

In this chapter, we propose a disparity estimation algorithm based on edge matching, propagation and interpolation. Because it does the propagation on edge map, so the object's boundary is much better than that by the algorithm proposed in [ML02]. The

interpolation method is based on the spatial correlation and color information, so it can interpolate the disparity value for the partially occlusion area. From our evaluation, it can be observed that the advantage of our algorithm can also estimate accurate disparity map using images without known camera geometry. However, there are two limitations of this algorithm. One is that for a region without matching feature points (usually this kind of region has no texture information) the algorithm cannot find the accurate disparity value inside such region. This also exists in other stereo matching algorithms. Another limitation is that the interpolation method is based on the color information. When foreground and background objects have similar color or objects have gradually changing color, the algorithm will not work as well. Disparity can be an important feature to segment video object from sequence. According to different applications, experimental results proved its efficiency.

Chapter 5
Conclusion and Future Work

5.1 Conclusion and Contribution

An automatic VOPs generation method for the support of object-based coding in the framework of MPEG-4 has been presented in this thesis that continuously separates moving objects in image frames through time evolution. The proposed method utilizes temporal information for localizing moving objects, and spatial information for the acquisition of precise object boundaries and semantic region partitions. Like most of the automatic segmentation algorithms, the proposed algorithm is based on motion information. This will be a limitation, if the object is stationary throughout the sequence. Disparity information is robust to motion fluctuation, even the person stays still for arbitrarily long period of time or when its different parts exhibit different motion characteristics. Disparity estimation is based on edge matching and spatial interpolation algorithm using color information, from which, it can be seen that all the pixels in the same region will have similar disparities. The boundaries of objects have no blocky artifacts, compared with other disparity estimation algorithms using block matching. Disparity information can combine with other features to segment objects according to different applications.

Some experiments have been carried out to test our proposed algorithms. Several MPEG-4 standard sequences were used to evaluate the proposed system. The results

showed that the proposed algorithm is capable of handling different kinds of videos. Stereoscopic image pairs' results showed the efficiency of the proposed disparity estimation algorithm.

The main contribution of our work can be summarized as:

- A clustering algorithm for video segmentation has been introduced. It is to segment a frame of video sequence into homogeneous regions with multiple features. In this algorithm, the cluster number and the weights of multiple features can be determined adaptively according to the characteristic of sequences.
- Edges are detected with weighted chrominance information, which can obtain more accurate edge map compared with the traditional method using only the luminance information.
- Robust object tracking and region segmentation are combined to improve segmentation accuracy and temporal coherence of moving objects.
- A disparity estimation algorithm is proposed based on edge matching, propagation and spatial interpolation.
- Disparity information is combined with other features to extract objects from video sequence.

5.2 Future work

There are still some shortcomings inherent in our VOP segmentation method, which must be overcome. More research is needed to be carried out in order to improve our algorithm.

1) How to reduce computation time. Current system is only good for off-line applications as the region partition is a time consuming process. To achieve a real-time computation target, many processes in the algorithm should be optimized.

2) Shadow effects, reflection, and noise might be incorrectly assigned to the foreground objects. The underlying problem is that the only motion we can observe is apparent motion, which is induced by temporal changes in the image intensity. Unfortunately, it is often very difficult to distinguish between changes due to true object motion and those due to noise, shadow effects, reflections, etc. New methods must be investigated to reduce the influence of shadow effects, reflections and noise.

3) In binary model tracking, once a wrong boundary is detected in one frame, it will persist throughout the sequence. The reason is that in our model update stage, all the pixels of the binary model are assumed to be exact. The model of the current frame is derived by choosing the set of edge pixels in the small distance of the model in the previous frame. If the background is cluttered, edge pixels belonging to the background regions may be chosen, which will result in wrong object boundaries.

4) There are some parameters in the algorithm which are now set by the user. It is preferred to determine them automatically according to the characteristic of the sequence.

5) Throughout the sequence, if there are several objects, they may interact with each other by merging, splitting or overlapping. The algorithm should be able to handle these kinds of interactions.

Reference

[AN79] A. N. Netravali and J. D. Robbins, "Motion compensated television coding: Part I," *Bell Syst. Tech. J.,* vol. 58, pp. 631-670, Mar. 1979.

[AN92] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *Visual Database Systems II,* E. Knuth and L. M. Wegner, Eds. Amsterdam, the Netherlands: Elsevier, pp. 113-127, 1992.

[AN05] A. N. Rajagopalan, R. Chellappa, and N. T. Koterba, "Background learning for robust face recognition with PCA in the presence of clutter," *IEEE Trans. on Image Processing*, vol.14, no.6, pp.832-843, June 2005.

[AV89] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, no. 5, pp. 490-498, May 1989.

[BE90] B. E. Dom, "Algorithm for Realtime Image Segmentation," *Vision'90* pp. 12-15, November 1990.

[BF99] B. Marcotegui, F. Zanoguera, P. Correia, R. Rosa, F. Marques, R. Mech, and M. Wollborn, "A video object generation tool allowing friendly user interaction," in *Proceedings of International Conference on Image Processing*, pp. 391-395, 1999.

[BH02] B. Huang, Y. Yang, Q. Wang and L. Wu, "Video segmentation based-on fuzzy clustering and multi-features, " *Int. Conf. Signal Processing*, , vol. 2, pp. 977-980. China, Aug. 2002

[BR02] R. V. Babu, K.R. Ramakrishnan, "Compressed domain motion segmentation for video object extraction", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.4. pp. 3788-3791, 2002.

[BS01] B. S. Manjunath, P. Salembier, and T. Sikora, "Introduction to MPEG-7: Multimedia Content Description Standard", New York: Wiley, 2001.

[CK05] C. Kim, "Segmenting a Low-Depth-of-Field Image Using Morphological Filters and Region Merging", *IEEE Trans. On Image Processing*, vol.14, no.10, October 2005

[CG98] C. Gu and M. C. Lee, "Semiautomatic segmentation and tracking of semantic video objects," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 572-584, 1998.

[CS97] C. Stiller, "Object-based estimation of dense motion fields," *IEEE Trans. Image Processing*, vol. 6, pp. 234–250, Feb. 1997.

[DC99] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *IEEE Trans. On Circuits and System for Video Technology*, vol. 9, pp. 551–564, June 1999.

[DP92] D. P. Huttenlocher, J. J. Noh, andW. J. Rucklidge, "Tracking Non-Rigid Objects in Complex Scenes," Dept. of Computer Science, Cornell Univ. Tech. Rep. 1320, 1992.

[DP93] D. P. Huttenlocher,G.A. Klanderman, andW. J. Rucklidge, "Comparing images using the Hausdorff distance," IEEE Trans. Pattern Anal. Machine Intell., vol. 15, pp. 850–863, Sept. 1993.

[DR84] D. R. Walker and K. R. Rao, "Improved pel-recursive motion compensation," *IEEE Trans. Comm.*, vol. COM-32, no. 10, pp. 1128-1134, October 1984.

[DR02] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," IJCV, April-June 2002.

[DW98] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 539–546, Sept. 1998.

[EC96] E. Chalom and V. Bove, "Segmentation of an image sequence using multi-dimensional image attributes", in *Proceedings of International Conference on Image Processing*, pp. 525-528, 1996.

[ET00] E. Tuncel and L. Onural, "Utilization of the recursive shortest spanning tree algorithm for video object segmentation by 2-D affine motion modeling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 776–781, Aug. 2000.

[FG98] J. G. Choi, M. Kim, J. Kwak, M. H. Lee, and C. Ahn, "User-assisted video object segmentation by multiple object tracking," *ISO/IECJTC1/SC29/WG11 MPEG98/m3349*, Tokyo, Japan, March 1998.

[GB86] G. Borgefors, "Distance transformation in digital image," *Computer Vision, Graphics, Image Processing*, vol. 34, pp. 344-371, 1986

[GD97] G. D. Borshukov,G. Bozdagi,Y. Altunbasak, and A. M. Tekalp, "Motion segmentation by multistage affine classification," *IEEE Trans. Image Processing,* vol. 6, pp. 1591–1594, Nov. 1997.

[HG90] H. Gharavi and M. Mills, "Blockmatching motion estimation algorithms-new results, " *IEEE Trans. Circuits and Systems*, vol. 37, no. 5, pp. 649-651, May 1990.

[HL03] H. Luo and A. Eleftheriadis, "Model-based segmentation and tracking of head-and shoulder video objects for real time multimedia services," *IEEE Trans. Multimedia*, vol.5, no.3, pp.379-389, Sep. 2003.

[HL06] Hongliang Li, and King N. Ngan, "Face Segmentation in Head-and-Shoulder Video Sequences Based on Facial Saliency Map", *IEEE ISCAS*2006.

[HT] http://mpeg.telecomitalialab.com/standards/mpeg-4/mpeg-4.htm

[http1] http://www.incx.nec.co.jp/imap-vision/library/wouter/median3.html

[HX04] H. F. Xu, A. A. Younis and M. R. Kabuka, "Automatic Moving Object Extraction for Content-Based Applications", *IEEE Trans. On Circuits and System for Video Technology*, vol. 14, no. 6, pp. 796–812, June 2004.

[IJ96] Ingemar J.Cox, Sunita L.Hingorani, Satish B.Rao, "A maximum likelihood stereo algorithm", *Computer Vision and Image Understanding*, vol.63, no.3, pp.542-567, May 1996.

[JC86] J. Canny, "A computational approach to edge detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, Nov. 1986.

[JN91] J. N. Driessen, L. Böröczky, and J. Biemond, "Pel-recursive motion field estimation from image sequences," *Journal of Visual Communication and Image Representation*, vol. 2, no. 3, pp. 259-280, September 1991.

[JG97] J. G. Choi, S.W. Lee, and S. D. Kim, "Spatio-temporal video segmentation using a joint similarity measure," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 279–286, Apr. 1997.

[JY98] J.Y.Tham, S. Ranganath, M.Ranganath, and A.A.Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, pp. 369-377, Aug. 1998.

[JZ01] J. Z. Wang, J. Li, R. M. Gray, and G. Wiederhold, "Unsupervised multiresolution segmentation for images with low depth of field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 85–90, Jan. 2001.

[KN99] K. N. Ngan, T. Meier and D. Chai, "Advanced Video Coding Principles and Techniques", Elsevier Science Publishers BV, ISBN 0-444-82667-X, August 1999.

[LC97] L. Chariglione, "MPEG and Multimedia Communications", *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 7, No. 1, pp. 5-18, February 1997.

[LF97] L. Garrido, F. Marques, M. Pardas, P. Salembier, and V. Vilaplana, "A hierarchical technique for image sequence analysis," in *Proc. Workshop Image Analysis for Multimedia Interactive Services (WIAMIS)*, Louvain-la-Neuve, Belgium, pp. 13–20, June 1997.

[LQ03] Q. Liu, M. Sun, R.J. Sclabassi, "An Application of MAP to Change Detection in Moving Video", Uncertainty Modeling and Analysis, ISUMA 2003. Fourth International Symposium on 2003, pp. 318 – 323, 2003.

[MA00] A. Marugame, A. Yamada and M. Ohta, "Focused Object Extraction with multiple cameras", *IEEE Trans. On Circuits and System for Video Technology*, vol. 10, no. 4, pp. 530–540, June 2000.

[MG99] Mohanmmed Ghanbari, *Video coding*, The Institution of Electrical Engineers London, United Kingdom, 1999.

[MK01] Mustapha Kardouchi, Janusz Konrad, Carlos Vazquez, "Estimation of largeamplitude motion and disparity fields: application to intermediate view reconstruction", Proceedings of SPIE, vol.4310, pp.340-351, VCIP'2001.

[ML02] M. Lhuillier and L. Quan, "Match propagation for image-based modeling and rendering," IEEE Trans. Pattern Anal. Machine Intell., Vol.24, No.8, pp.1140-1146, Aug. 2002.

[ML04] M. Lievin, and F. Luthon, "Nonlinear color space and spatiotemporal MRF for hierarchical segmentation of face features in video," IEEE Trans. Image Processing, vol.13, no.1, pp.63-71, Jan. 2004.

[MM94] M. M. Chang, M. I. Sezan, and A. M. Tekalp, "An algorithm for simultaneous motion estimation and scene segmentation," in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, vol. V, Adelaide, Australia, Apr. 1994, pp. 221–224.

[MM97] M. M. Chang, A. M. Tekalp, and M. I. Sezan, "Simultaneous motion estimation and segmentation," IEEE Trans. Image Processing, vol. 6, pp. 1326–1333, Sept. 1997.

[MP93] MPEG-1 Video Group, "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media up to about 1.5 Mbit/s: Part 2 - Video," *ISO/IEC 11172-2, International Standard*, 1993.

[MP94] M. Pardas and P. Salembier, "3-D morphological segmentation and motion estimation for image sequences," Signal Process., vol. 38, no. 1, pp. 31–43, Sept. 1994.

[MP95] MPEG-2 Video Group, "Information Technology - Generic Coding of Moving Pictures and Associated Audio: Part 2 - Video," *ISO/IEC 13818-2, International Standard*, 1995.

[MP98] MPEG-4 Video Group, "Generic Coding of Audio-Visual Objects: Part 2 - Visual," *ISO/IEC JTC1/SC29/WG11 N1902, FDIS of ISO/IEC 14496-2*, Atlantic City, November 1998.

[ND00] N. D. Doulamis, A. D. Doulamis,Y. S. Avrithis, K. S. Ntalianis and S. D. Kollias, "Efficient Summarization of Stereoscopic Video Sequences", *IEEE Trans. On Circuits and System for Video Technology*, vol. 10, no. 4, pp. 501–517, June 2000.

[NH04] N. Habili, C. C. Lin and A. Moini, "Segmentation of the Face and Hands in Sign Language Video Sequences Using Color and Motion Cues," *IEEE Trans. On Circuits and System for Video Technology*, vol. 8, pp. 1086–1097, August 2004.

[NN03] N. Nariman and K. N. Ngan, "Automatic Multi-cue VOP Extraction for MPEG-4," *Picture Coding Symposium 2003*, Saint Malo - France, Apr. 2003.

[NP00] N. Paragios and R. Deriche, "Geodesic active contours and level sets for the detection and tracking of moving objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 266–280, Mar. 2000.

[OE02] E. P. Ong, B. J. Tye, W. S. Lin and M. Etoh, "An efficient video object segmentation scheme", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002, vol. 4, pp. IV3361-3364

[PJ87] P. J. Rousseeuw and A. M. Leroy, Robust Regression and Outliner Detection, John Wiley & Sons, New York, NY, 1987.

[PS94] P. Salembier and M. Paradas, "Hierarchical morphological segmentation for image sequence coding," *IEEE Trans. Image Processing,* vol. 3, pp. 639–651, Sept. 1994.

[PS96] Ph. Schroeter, "Unsupervised two-dimensional three-dimensional image segmentation, " Ph.D dissertation, Swiss Federal Inst. Technol., Lausanne, Switzerland, 1996.

[PS99] P. Salembier, F. Marqués, "Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, no.8, December 1999.

[RC98] R. Castagno, T. Ebrahimi, and M. Kunt, "Video segmentation based on multiple features of interactive multimedia applications," *IEEE Trans. On Circuits and System for Video Technology,* vol. 8, pp. 562-571, September 1998.

[RS98] R. Schafer, "MPEG-4: A Multimedia Compression Standard for Interactive Applications and Services", *Electronics and Communication Engineering*, December1998.

[SA98] S. Arora, P. Raghavan, and S. Rao, "Approximation schemes for Euclidean k-medians and related problems," *Proc. 30th Annual ACM Symposium on Theory of Computing*, pp. 106-113, 1998.

[SC98] S.Colonnese and G.Russo, "User interaction modes in semi-automatic segmentation: Development of a flexilbe graphical user interface in Jave," *ISO/IEC JTC1/SC29/WG11 MPEG98/m3320*, Tokyo, Japan, March 1998.

[SF01] S.-F. Chang, T. Sikora, and A. Puri, "Overview of MPEG-7 Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, pp. 688–695, June 2001.

[SM97] S.M. Smith and M. Brady, "SUSAN - a new approach to low level image processing", *International Journal of ComputerVision*, Vol. 23(1), 45-78, 1997.

[ST97] T Sikora, "MPEG-4 very low bit rate video," *Processings of IEEE ISCAS Conference*, Hong Kong, June 1997.

[SY02] S. -Y. Chien, S.-Y. Ma, and L.-G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 12, pp. 577–586, Jul. 2002.

[SY03] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Predictive Watershed: A Fast Watershed Algorithm for Video Segmentation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, pp. 453–461, May 2003.

[SZ05] S. Z. Li, X. Lu, X. Hou, X. Peng, and Q. Cheng, "Learning multiview face subspaces and facial pose estimation using independent component analysis," *IEEE Trans. on Image Processing*, vol.14, no.6, pp.705-712, June 2005.

[TH86] T. H. Wonnacott, Regression, a Second Course in Statistics, Wiley, New York, NY, 1986.

[TM98] T. Meier and K. N. Ngan, "Automatic segmentation of moving objects for video object plane generation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 525–538, Sept. 1998.

[TS97] T. Sikora, "The MPEG-4 Video Standard Verification Model", *IEEE Trans. of Circuits and Systems for Video Technology*, Vol.7, February 1997.

[TM99] T. Meier and K. N. Ngan, "Video Segmentation for Content-Based Coding," *IEEE Trans. On Circuits and System for Video Technology*, vol. 9, no. 8, December 1999.

[VM04] Vasileios Mezaris, Ioannis Kompatsiaris and, Michael G. Strintzis,"Video Object Segmentation Using Bayes-Based Temporal Tracking and Trajectory-Based Region Merging", *IEEE Trans. On Circuits and System for Video Technology*, vol. 14, no. 5, pp. 782–795, June 2004.

[WW00] Woontack Woo, Antonio Ortega, "Overlapped block disparity compensation with adaptive windows for stereo image coding", *IEEE trans. On Circuits and Systems for Video Technology,* vol. 10, no.2 March 2000.

[WJ01] W.J. Heng and K.N. Ngan, "An object-based shot boundary detection using edge tracing and tracking," *Journal of Visual Communications and Image Representation,* Academic Press, U.S.A., Vol. 12, No. 3, pp.217-239, September 2001.

[XP00] X. Marichal and P.Villegas, "Objective evaluation of segmentation masks in video sequences," in Proceedings of X European Signal Processing Conference (EUSIPCP), (Tampere, Finland), pp. 2193-2196, 2000

[YD01] Y. Deng and B.S.Manjunath, "Unsupervised segmentation of Color-Texture Regions in Images and Video", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 23, pp 800-810, Aug. 2001.

[YH94] Y. Hu and T. J. Dennis, "Textured image segmentation by context enhanced clustering," *IEE Proc. –Vision, Image and Signal Processing*, vol. 141, pp.413-421, Dec. 1994.

[YP05] Yu-Pao Tsai, Chih-Chuan Lai, Yi-Ping Hung, and Zen-Chung Shih, "A Bayesian Approach to Video Object Segmentation via Merging 3-DWatershed Volumes", *IEEE Trans. On Circuits and System for Video Technology*, vol. 15, no. 1, pp. 175–180, January 2005.

[YW01] Y. Wang, Jörn Ostermann, and Y. Q. Zhang, *Digital Video Processing and Communications,* Prentice-Hall, 2001.

[YZ00] Y. Zhong, A. K. Jain, and M.-P. Dubuisson-Jolley, "Object tracking using deformable templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 544–549, May 2000.

[ZZ95] Zhang, Z.& Deriche, R. & Faugeras, O & Luong, Q, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry", *Artificial Intelligence Journal*, 78:87-119, October 1995.